Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Module Counterfactuals for Attributing LLM Serving Scheduler Effects

Soroush Vahidi

sv96@njit.edu

New Jersey Institute of Technology

Research Article

Keywords: LLM serving · Scheduling · Module interventions · Counterfactual replay · Performance evaluation · Benchmarking

Posted Date: November 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11569472/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

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

Module Counterfactuals for Attributing LLM Serving Scheduler Effects

Soroush Vahidi

Abstract An unchanged aggregate score can hide a scheduler component's effects. We specify module interventions for large language model serving that pair complete runs on matched workloads and compare recommendations at identical simulated states. Beyond which component is replaced, the contract declares observations and units distinguishing opportunity, recommendation, executed action, request outcome and aggregate outcome. In controlled simulations, replacing the input-prompt component changes completion timing or deadline status for about 59 percent of requests while leaving weighted deadline attainment unchanged. Across three selected component pairs, about 99 percent of recommendation disagreements disappear before admission; additional pairs show that erasure depends on the component and surrounding scheduler. Composition interactions, workload dependence and endpoint sensitivity limit transfer of aggregate conclusions. Thus, an aggregate null does not establish inactivity, and a changed recommendation does not establish a changed action or performance benefit. On seven constructed cases, the protocol and equally instrumented ablation produce identical diagnoses. The contribution is an explicit experimental specification and scoped empirical evidence, not diagnostic superiority or a new causal estimator. Conclusions remain model-conditional; native fidelity and identical-input serving-engine confirmation remain unestablished.

Keywords LLM serving · Scheduling · Module interventions · Counterfactual replay · Performance evaluation · Benchmarking

Acknowledgments: The author thanks Professor Ioannis Koutis for his support and Anders Borum for complimentary lifetime access to Secure ShellFish, used for remote access and research development. Portions of this research used Wulver, provided by the Advanced Research Computing Services (ARCS) team at the New Jersey Institute of Technology (NJIT).

Soroush Vahidi (corresponding author)
New Jersey Institute of Technology, Newark, NJ, USA
E-mail: sv96@njit.edu
ORCID: 0000-0003-1934-6282

1 Introduction

An unchanged deadline-attainment score can conceal different scheduler behaviors: no competing requests, a changed ranking erased by resource filtering, or changed completion times without deadline crossings. We ask: *why can replacing a scheduler module change behavior while leaving aggregate performance unchanged?*

Large language model (LLM) serving combines prefill, which processes input prompts, with decoding, which generates tokens. Iteration-level batching and paged key–value (KV) memory support scheduling and reusable attention state [10, 23]; chunked prefill and fairness further change service allocation [2, 16]. A module's consequences depend on this composition and its workload.

We distinguish opportunity, recommendation, executed action, request consequence and aggregate consequence (Fig. 1). A typed substitution replaces one module while holding the surrounding scheduler fixed. Matched full trajectories measure persistent replacement effects; same-state comparisons follow alternative recommendations through fixed downstream logic at cloned base-visited states. Request matching links trajectory outcomes to individuals. These are complementary observations, not a single longitudinal mediation study.

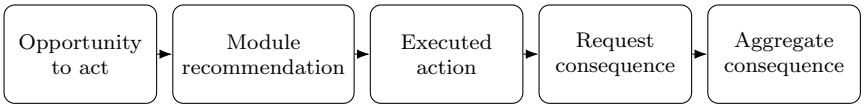


Fig. 1 Diagnostic observations needed to interpret an aggregate null. Same-state comparisons test recommendation-to-action survival; paired full trajectories supply request and aggregate consequences. Arrows organize questions, not scheduler execution order or an identified mediation model. Different campaigns examine different links

The contribution is a diagnostic contract that extends the specification of *what to replace* with *what to observe and compare*: typed boundaries, matched workloads, cloned-state recommendations, downstream actions and matched-request outcomes, each with a declared unit and denominator. It combines established ablation, paired simulation and factorial contrasts; equivalent instrumentation recovers the same observations in all seven constructed comparator cases (Sect. 4.2). The empirical contribution is to show how erasure, request changes hidden by aggregate nulls, and composition, workload and endpoint dependence qualify a module-effect claim.

RQ1 asks when modules have opportunity to act; **RQ2**, where recommendation differences disappear or escape an aggregate endpoint; **RQ3**, how composition, workload and endpoint limit transfer; and **RQ4**, what reconstructed mechanisms and external checks validate. The evidence concerns simulated mechanisms, not identified production effects.

2 Related Work

2.1 Ablation, sensitivity and interaction experiments

Configuration ablation replaces settings along a path between algorithms [8]; replacement attribution is established. Our contract specifies typed scheduler boundaries and local, action and matched-request observations. Equivalent instrumentation can supply them (Sect. 4.2); we claim no new replacement method.

Variance-based sensitivity analysis apportions output variability over an input distribution [17]. Our fixed-workload paired effects and base-state local erasure are not variance shares. Factorial contrasts and common random numbers are established tools for interactions and paired simulation [11, Sections 11.2 and 12.2]. We apply them to how one module's context changes another's effect.

2.2 Causal profiling and counterfactual replay

Coz tests virtual code speedups and validates optimization guidance on real applications [6]. We test recommendation substitutions and downstream survival, without equivalent deployment validation. CausalSim learns causal dynamics from randomized trials to address algorithm-dependent trace bias [3]. We perform no such identification: replayed workload fields are specified inputs to a reconstruction. Its counterfactual contrasts do not remove production selection bias or identify deployment effects.

2.3 LLM-serving systems and evaluation

Orca's iteration-level scheduling, vLLM's paged KV memory and Sarathi-Serve's chunked prefill establish complementary serving mechanisms [2, 10, 23]. Their composition makes request ordering, locality and resource allocation interdependent. VTC accounts for token service to enforce fairness [16]; Fu et al. predict relative output-length ranks to guide request scheduling [9]. Preble co-optimizes cached prompt-state reuse and distributed computation load balancing [18]. Cao et al.'s 2025 preprint *Locality-aware Fair Scheduling in LLM Serving* (arXiv:2501.14312) studies prefix locality, fairness and distributed load balancing together.

Architectural choices also change where scheduling acts. Splitwise and DistServe separate prefill and decode; DistServe couples phase-specific resource allocation to latency constraints [14, 24]. Mooncake uses a KV-centric architecture [15], while Llumnix reschedules live requests and their state across instances [20]. Laser studies layer-level scheduling for multiple SLOs [13]; CrossServe addresses cross-layer scheduling for SLO optimization in multi-tenant serving [12]. These mechanisms motivate studying composition; none

is reimplemented or used as an experimental baseline here. Our contribution specifies module-attribution observations, rather than a learned ranking scheduler or serving optimizer.

The SCORPIO preprint by Tang et al. (2025), *Serving the Right Requests at the Right Time for Heterogeneous SLOs in LLM Inference* (arXiv:2505.23022v1) motivates SLO-aware admission and batching. Our Scorpio-labeled components are project reconstructions; their specific provenance and compatibility limits are evaluated in Sect. 7.2.

Vidur combines profiling and prediction for performance estimation and configuration exploration [1]. LLMServingSim 2.0 models runtime interactions among serving decisions, heterogeneous hardware and disaggregated infrastructure; its authors report comparisons against real deployments [5]. Our reconstruction inherits neither framework's hardware validation. Simulation can supply detailed observations, but model-conditional counterfactual conclusions, fidelity to native systems and production causal identification remain distinct. Online Resource 1, Table S3 compares the attribution questions.

2.4 Related manuscript and shared evidence

Soroush Vahidi's SSRN preprint *When Does LLM-Serving Scheduler Adaptation Matter? Action Opportunity and Causal Headroom in Production-Derived Replay* (2026; `https://doi.org/10.2139/ssrn.7584825`) studies whole-policy opportunity and one-step latency headroom under a fixed continuation policy. Here, local comparisons stop at admitted-set equality, while trajectory comparisons retain a module substitution throughout replay. The projects share simulator infrastructure, reference-conditional counterfactual reasoning and some Azure/BurstGPT records; their datasets are not independent. Record-level overlap and source-version qualifications are given in Online Resource 1, Sect. S23. The related findings are not independent validation of the present protocol.

3 Problem Setting and Counterfactual Estimands

3.1 Composed schedulers and admissible substitutions

A scheduler policy is represented by typed modules

$$p = (m_{\mathrm{adm}}, m_{\mathrm{pri}}, m_{\mathrm{pre}}, m_{\mathrm{kv}}, m_{\mathrm{fair}}). \tag{1}$$

The slots are admission, priority, prefill handling, KV guard and fairness aging. The tuple specifies slots, not their execution order. A *base* policy b is the experimental control. A donor policy d supplies a module d_s for slot s; $b[s \leftarrow d_s]$ denotes that substitution with the remaining modules fixed. The donor's full policy is not the treatment arm, and a literature scheduler is not a control merely because its name appears in the inventory.

A workload instance w fixes arrivals, request sizes, service-level objectives (SLOs), priority fields, replay transformations and a random-seed namespace. At a scheduler state x, a module produces a recommendation $q_s(x)$. The surrounding scheduler maps this recommendation to an executable action $a(x)$ through filtering, ranking and resource constraints. A typed interface makes this boundary inspectable; it does not by itself prove that a donor's state assumptions or semantics are valid in the base context.

We distinguish three requirements. *Execution compatibility* means the replacement runs. *Semantic compatibility* means the available state and decision object have the intended meaning. *Intervention isolation* means only the declared boundary changes. Missing companion state, different time units, hidden mutable history or a silently inactive implementation can break interpretability even when the code executes. The protocol in Sect. 4 records these requirements and preserves failures rather than treating runtime acceptance as validation.

3.2 Trajectory effects and endpoint definitions

Full-trajectory comparison starts the base and substituted compositions from the same initial condition and replays the same workload. With common random numbers (shared random inputs across arms) [11, Section 11.2], let $R(p, w)$ denote the aggregate outcome of policy p on workload w. Intervention $i = (b, s, d_s)$ has paired aggregate contrast

$$\Delta_i(w) = R\big(b[s \leftarrow d_s], w\big) - R(b, w). \tag{2}$$

This is the effect of the specified replacement within the simulated model and workload. It includes any subsequent state changes along the two trajectories. It neither estimates a universal contribution of slot s nor identifies how a deployed production system would respond.

Our primary endpoint is arrival-normalized weighted goodput (ANWG). Let $\mathcal{A}$ be all arrivals, $\mathcal{C}$ the completed requests, t_i the completion time, d_i the absolute deadline and $w_i > 0$ the priority weight (default 1 for non-positive input priority). The indicator $\mathbf{1}[\cdot]$ equals one when its condition holds and zero otherwise. Then

$$R = \mathrm{ANWG} = \frac{\sum_{i \in \mathcal{C}} w_i \mathbf{1}[t_i \le d_i]}{\sum_{j \in \mathcal{A}} w_j}. \tag{3}$$

Dropped or uncompleted arrivals contribute to the denominator but not the numerator. A positive Δ_i is a benefit on this scale. Equal ANWG does not imply equal request completion times or equal service allocation.

Secondary endpoints are specified before interpreting the primary nulls. Weighted completion goodput is the weighted fraction of arrivals that complete. Mean soft deadline credit averages $\exp(-\max(0, t_i - d_i)/S_i)$ over completion records, with $S_i = \max(d_i - \mathrm{arrival}_i, \varepsilon)$, where $\varepsilon > 0$ is a numerical floor in simulator time units. Weighted mean tardiness averages $\max(0, t_i - d_i)$ with

weights w_i over those records; weighted normalized tardiness first divides by S_i. Lower is better for the two tardiness measures. Request-level comparisons match request identifiers and record changed completion time, SLO status and unmatched completions. These endpoints describe the simulator; they are not interchangeable with measured hardware latency.

3.3 Same-state disagreement and downstream erasure

A trajectory difference cannot generally be localized by comparing arbitrary later steps: the two schedulers may then face different queues and resource states. Instead, the local comparison clones a state x on the base trajectory and holds all other logic fixed. It evaluates the base and donor recommendations and their downstream actions at that common state.

Structural opportunity is a performance-blind condition under which a module can affect a choice, such as at least two priority candidates. *Actionable disagreement* means the selected base and donor recommendations differ on the same candidate set. In the local experiments the final action is the admitted-request-ID *set*; equality does not establish equal ordering or every aspect of execution. A disagreement is *erased* (overridden) when the two recommendations produce the same final admitted set after downstream composition. The first applicable stage in the implemented attribution taxonomy assigns the cause. This is a conditional diagnostic classification, not identification of independent causal mediators.

Let C be the set of evaluated pair–window cells. For $c \in C$, let n_c count actionable events and e_c count those erased. We report two different estimands:

$$\widehat{E}_{\text{event}} = \frac{\sum_c e_c}{\sum_c n_c}, \qquad \widehat{E}_{\text{cell}} = \frac{1}{|C_+|} \sum_{c \in C_+} \frac{e_c}{n_c}, \quad C_+ = \{c \in C : n_c > 0\}. \tag{4}$$

The first weights cells by event count; the second gives equal weight to positive-denominator cells. A cell with no actionable event has no defined rate, not a zero rate. Events in a window share the workload and scheduler history and are not independent samples. Uncertainty therefore resamples complete windows (Sect. 4.5).

A high erasure fraction at base-visited states does not establish equivalent full trajectories. A surviving difference can change which states are visited later; local equality also leaves unmeasured state and action representations outside its scope. The local analysis conditions on the selected base policy, module pairs and state distribution.

Saved pilot cases in Online Resource 1, Sect. S25, show how these observations distinguish aggregate nulls: both-fit admission erases a ranking difference, whereas reordered service changes two request outcomes without changing ANWG.

4 Methodology and Experimental Design

4.1 Intervention construction and observation protocol

Each intervention records the base policy, donor policy, module type, donor implementation and immutable identifier. The control is the unchanged base composition; the single treatment replaces one declared slot. We retain the same workload, initial state and random-seed namespace across paired arms. The protocol proceeds as follows:

1. Specify the slot's input state and decision object. Check that the donor has an executable branch, that required state is available, and that the remaining composition is fixed. Record unresolved semantic assumptions.
2. Run the paired full trajectories and match requests across arms. Save aggregate outcomes and request consequences rather than inferring one from the other.
3. Record mechanism-specific opportunity and action diagnostics. For a local erasure question, clone base-visited states and compare recommendations through the same downstream logic, keeping those observations separate from trajectory differences.
4. Aggregate with the declared denominator and sampling unit. Keep implementation versions and evidence strata separate; report absent opportunity, null outcomes and failed validity checks alongside active effects.

A graphics processing unit (GPU) executes model computation. Laxity is the remaining deadline slack after estimated service; a guard is a condition that filters eligibility. Fairness aging increases scheduling priority with waiting time. Table 1 defines the implemented interfaces. Specification-level isolation checks compare the slots changed by single and paired interventions. They are necessary but weaker than a full state-and-mutation contract. The KV-specific failures are evaluated in Sect. 7.2. A donor returning the base recommendation can be a valid observed null; an inactive implementation bug is instead excluded from corrected-effect claims.

Priority probes record candidate-set size and rank disagreement; a nontrivial candidate set does not guarantee a different top choice. Prefill probes record pressure above the frozen threshold 0.65, filter disagreement and paired action divergence. Fairness probes record promotion opportunity under the frozen aging rule. These probes locate the opportunity presented by the model; they are not performance labels used to select successful runs. The diagnostic ordering from opportunity to consequences should not be confused with the scheduler's execution ordering below. Separate campaigns probe these links; they do not form a single longitudinal mediation study.

4.2 Equally instrumented ablation comparator

A seven-case processor-only pilot checks the distinction between obtaining an aggregate contrast and diagnosing its mechanism. An instrumented abla-

Table 1 Typed interfaces in the reconstructed scheduler. These are model boundaries, not a certificate of native donor fidelity.

Slot	State observed	Decision object
Admission	Waiting queue, laxity, GPU/KV headroom	Filtered candidate set
Prefill	Queue pressure and token lengths	Guard/chunk eligibility filter
Priority	Filtered candidates, time and SLO fields	Ranked request order
Fairness aging	Waiting time and service estimate	Promotion-adjusted rank key
KV guard	GPU KV occupancy and request sizes	Reserve/placement eligibility

tion uses generic relational queries; the protocol implementation uses explicit stage predicates. Both receive identical state records, cloned local branches and matched request completions, with the same acquisition budget. They produce identical diagnoses in all seven cases. An aggregate-only comparison is allowed to remain ambiguous and is not scored as incorrect.

The 56 full trajectories include logger-off/on controls and a fresh-process determinism repeat, not 56 independent cases. A separate schedule/arithmetic checker verifies the fixtures without importing either analyzer or the simulator. This supports the implemented observation contracts, including negative and null cases, but not general diagnostic accuracy, computational advantage or superiority to equally instrumented ablation. The cases were constructed with known schedules and the implementation was not blinded. Online Resource 1, Sect. S11 gives all cases and accounting.

4.3 Four-arm module interactions

To test whether context changes a module effect, evaluate base, module A only, module B only and both on the same workload. Our unscaled factorial contrast uses $R_\emptyset$, R_A, R_B and R_{AB} for their respective outcomes on the same workload:

$$I(A,B) = R_{AB} - R_A - R_B + R_\emptyset = \Delta(A+B) - \Delta(A) - \Delta(B). \tag{5}$$

This difference in differences is twice the conventional two-factor interaction effect [11, Section 12.2, p. 634]. All four arms remain paired. We estimate it only for declared joint designs, not by subtracting unrelated campaign means. Its meaning depends on the endpoint scale: $I = 0$ is additivity on ANWG,

not absence of all behavioral interaction. A positive I can indicate less-than-additive harm when both isolated effects are negative; it need not indicate benefit relative to the base.

4.4 Simulator semantics and fidelity

The discrete-event simulator advances scheduling and service in steps. The inspected pinned reconstructed-policy versions execute admission filtering, prefill filtering, priority ranking, fairness promotion, then placement under sequence-capacity and KV-token constraints. Service subsequently advances and completed requests drain. Supported placement paths invoke the reserve guard. Thus prefill precedes priority in these versions; the slot tuple is not an alternative execution-order claim.

The local-erasure source and controlled-campaign snapshots support this ordering. Historical deployment evidence consists of recorded commits, manifest hashes and integrity checks; the current checkout is not assigned retrospectively as the executable for every run. Exact execution-level attestation remains incomplete for older KV outputs and some replay records. The service model uses calibrated proxies, but the historical decode path does not fully consume calibrated decode timing. Numerical differences in simulator time therefore cannot be treated as calibrated GPU-latency gains. Policy reconstruction checks establish consistency with available pinned code, not equivalence to production vLLM or to every named native algorithm. Online Resource 1, Sect. S22 records the relevant provenance.

4.5 Statistical units and interpretation

The controlled priority and prefill experiments each contain three workload conditions with ten independent generation seeds per condition. Multiple interventions evaluated on one workload are paired observations, not additional independent workloads. The historical 95% percentile bootstrap intervals use $B = 2000$ draws of workload-instance summaries [7]. They are pointwise confidence intervals (CIs). Benjamini–Hochberg (BH) false discovery rate (FDR) adjustment at $q = 0.10$ applies to the frozen testing families, not to every interval in a figure [4]. Dependence among contrasts means nominal FDR control is not established for these families.

Zero bootstrap tail hits do not imply $p = 0$. At $B = 2000$, two-sided resolution is 0.001; a post-hoc plus-one correction retains the 17/28 priority and 0/4 prefill classifications (Online Resource 1, Sect. S27). These are bootstrap diagnostics, not exact randomization tests. The practical ANWG threshold is 0.002.

The post-hoc erasure analysis resamples eight complete windows within each of seven fixed sources, preserving all pairs and three seeds from a window ($B = 10{,}000$). Zero-denominator cells remain in the sampling frame but

supply no defined cell rate; an all-zero draw would be recorded as undefined and omitted. None occurred. The event and positive-cell estimands in Eq. 4 are recomputed separately in each draw. The intervals assume exchangeable windows within source; disjoint source-row ranges do not eliminate possible temporal dependence.

Public replay instead has only three deliberately selected windows per source, each evaluated at three rates. Its corrected bootstrap retains every rate within a window, resamples windows within seven fixed source strata and recomputes source centering per draw. These percentile ranges are conditional *sensitivity* summaries, not calibrated population confidence intervals. The earlier config-ID procedure incorrectly separated rates and is withdrawn. Both corrected implementations derive portable seeds with the 256-bit Secure Hash Algorithm (SHA-256). Observed exact zeros are finite-design findings, not equivalence tests; the 10^{-12} metric-classification tolerance is numerical, not a practical-effect or significance threshold.

4.6 Workloads, selection and evidence strata

The experiments vary opportunity and pressure to expose different parts of the attribution problem. Table 2 gives the controlled conditions: low competition versus active priority competition, and inactive versus sustained prefill pressure, with separate joint-opportunity conditions for interactions. All six conditions use mean output length 220 tokens, burst multiplier 3.0, burst-duration fraction 0.20 and ten independent seeds. The priority experiments evaluate 28 single substitutions; the prefill experiments evaluate four. The labels P0–P2 and F0–F2 identify these fixed experimental conditions.

Table 2 Controlled workload conditions. ρ is target utilization; capacity is the GPU sequence limit; prompt is mean input tokens; SLO multiplier scales deadline slack. KV budgets are in tokens.

Condition	ρ	Capacity	Prompt	SLO	KV	Aging
P0: no competition	0.05	4	512	0.10	18k	off
P1: competition	0.72	4	512	0.10	18k	off
P2: joint opportunity	0.80	4	512	0.10	18k	on
F0: inactive prefill	0.05	8	512	2.00	18k	off
F1: active prefill	0.92	4	3072	0.75	18k	off
F2: joint opportunity	0.92	4	1600	0.10	16k	on

The local-erasure follow-up selects three pairs from an inventory of 38 implemented benchmark-eligible pairs: an admission-control base with priority from the KV-constrained policy, prefill from adaptive chunked prefill, or fairness from aging priority. The selection was fixed before this follow-up, using

prior mechanistic evidence and feasibility. It is not random selection from all pairs, nor outcome-free selection across the research history. The remaining 35 pairs were not assessed for this endpoint. The design contains 56 source windows, eight per source, with three seeds per window; three pairs give 168 pair–window cells after aggregating seeds.

Generalization uses 45 fresh synthetic workload instances and a separate public replay design with 21 source windows at rates 0.5, 1 and 2. The seven source strata are Azure 2023/2024 code and conversation, BurstGPT, and Mooncake conversation and ToolAgent traces [14, 15, 19, 22]. Each public window contains 72 records: 1,512 distinct source records become 4,536 replay appearances in 63 dependent window–rate instances. A prospective opportunity follow-up uses 140 new windows at three replay rates (420 window–rate cells), disjoint from those 21 windows. Its primary inference uses the 140 unit-rate windows after seed aggregation; the other rates are prespecified sensitivities.

Only fields present upstream are observed: arrivals, input/output token lengths and, where available, Mooncake cache-hash structure. Priorities, deadlines/SLOs, replay rates, module interventions and their outcomes are assigned or simulated. Shared-data overlap is disclosed in Sect. 2; trace provenance does not change the model-conditional estimands.

Controlled and prospective protocols were preregistered; held-out designs use fresh seeds. Implementation corrections and uncertainty repairs are post-hoc. Reused outcomes are not independent replications, and fresh workloads in the same simulator do not constitute independent-system validation. Follow-ups can depend on earlier findings even when their own analysis is prespecified. Online Resource 1 retains protocols and seed details.

The additional erasure extension uses two different base/donor pairs on the 21 public-replay windows at unit rate, with three deterministic seed checks per pair and window. Dynamic-fidelity campaigns instead compare donor and transplant recommendations on sampled cloned states: 57 candidates and four prefill candidates respectively, each on 63 window–rate configurations and three seed settings. Their populations are reported separately in Sects. 5.2 and 7.1; they do not enlarge the original erasure denominator.

4.7 Substantive AI assistance

The author used OpenAI Codex for coding and ChatGPT (OpenAI) and Gemini (Google) for writing. The project disclosure also records Cursor (Anysphere), Claude (Anthropic) and Perplexity AI across software, literature review, workflow design, analysis checks and drafting; individual roles and model versions are not fully recorded. Journal revisions used Codex for claim and provenance audits, bounded tests, saved-data reanalysis, figures, substantive editing and release checks. AI outputs were not experimental evidence. Figures use Matplotlib or TikZ code with saved observations or author-developed diagrams; no generative image model produced empirical content. The author has personally reviewed the AI-assisted scientific revisions, analyses, citations and

manuscript text and retains responsibility for the work. This personal review is distinct from independent external verification.

5 Experimental Results

The results follow the diagnostic chain in Fig. 1, with separate experiments for its different links. Unless stated otherwise, controlled intervals are pointwise 95% workload-instance bootstrap intervals and adjusted classifications use the frozen BH $q = 0.10$ families.

5.1 Opportunity and decision changes

Tight deadlines alone do not expose a priority rule. Under P0, there is SLO pressure but no candidate competition; all 28 priority substitutions have observed zero decision divergence and zero ANWG change. Under P1, the fraction of steps with at least two candidates is approximately 0.35, and rank disagreement is approximately 0.27. Final decision divergence is much lower: 0.0123 $[0.0094, 0.0154]$. Candidate competition is therefore an opportunity for influence, not a guarantee that a ranking difference reaches the action. Figure 2 contrasts opportunity and outcome levels.

Prefill has a related activation boundary. In the inactive control F0, action, request and ANWG differences are all observed zeros. In F1, pressure exceeds 0.65 for a mean 0.553 fraction of steps (minimum 0.318), and paired action divergence is 0.0159 $[0.0132, 0.0188]$. The F0–F1 divergence contrast is FDR-significant. These controlled contrasts establish activation in the tested settings; they do not establish a universal relationship between structural opportunity and effects on other workloads.

5.2 Downstream erasure of recommendation differences

After full trajectories diverge, a later difference in actions can reflect different states as well as different modules. Approximately 63.7% of actionable trajectory steps were unclassifiable (65.1% among overrides): the same 232,011 unknown classifications divide by 364,302 actionable steps or 356,577 overrides, respectively. Conditioning on the same base-visited state removes this particular ambiguity within the implemented local taxonomy. It changes the estimand rather than repairing the trajectories into an equivalent comparison.

The three selected pairs produce 340,548 actionable events, of which 338,241 are erased before the final admitted set. Among 168 pair–window cells, 72 have positive denominators and 96 have no actionable disagreement. Positive support occurs in 40 of the 56 windows. Each cell aggregates three seeds; Table 3 separates pair-specific support and weighting. Events are not independent replications.

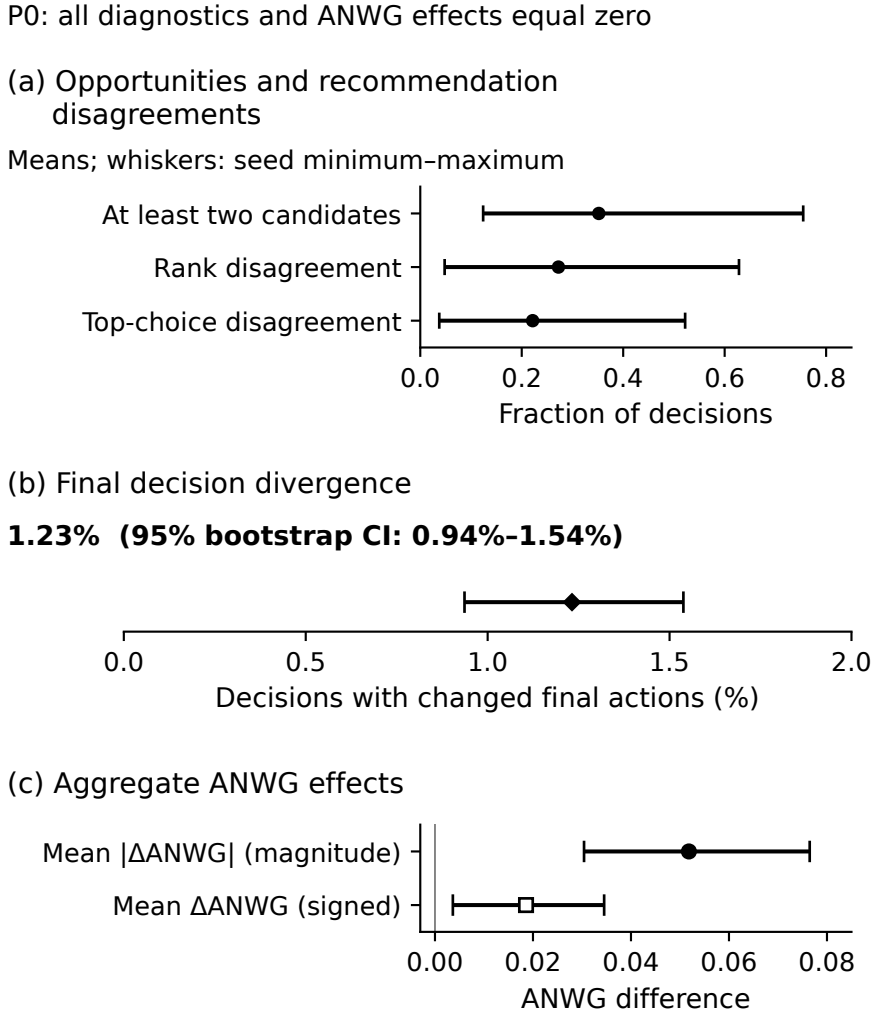


Fig. 2 Priority-module substitutions under P1 competition (10 independent workload seeds per cell); all P0 diagnostics and effects are zero. (a) Diagnostic means with seed minimum–maximum whiskers. (b) Fraction of decisions whose final actions change under the tested substitutions: 1.23%, pointwise 95% bootstrap CI 0.94%–1.54%. Its percentage axis is separate: the upper diagnostics measure opportunity or recommendation disagreement, not final action changes. (c) Mean absolute ΔANWG measures magnitude; signed mean ΔANWG measures benefit (positive) or harm (negative). Each has its own pointwise 95% bootstrap CI. Fixed interventions are averaged within workload before bootstrapping

Table 3 Original erasure study, by selected pair: 56 windows per pair, three seeds aggregated per cell. Event rates and equal-weight positive-cell means are descriptive; no pair-specific interval is implied.

Module	Positive / zero	Erased / actionable	Event (%)	Cell (%)
Priority	39 / 17	237390 / 238863	99.38	93.83
Prefill	27 / 29	97812 / 98550	99.25	99.23
Fairness	6 / 50	3039 / 3135	96.94	96.54

The pooled event-weighted estimate is 99.3226%, with post-hoc interval [98.80%, 99.58%]. The equal-weight positive-cell mean is 96.0806%, with its own interval [93.23%, 98.01%]. These are the two estimands in Eq. 4, computed by the complete-window resampling in Sect. 4.5. The difference reflects event concentration across cells. Zero-denominator cells remain in sampled windows without being assigned zero rates.

Erasure is heterogeneous. The positive-cell median is 99.1%, with interquartile range 96.7%–99.8% and range 0–100%; the zero-rate cell has only three actionable events. The historical quartiles select the sorted value at index $\lfloor q(n-1) \rfloor$ (zero-based), with $n = 72$ and $q = 0.25, 0.75$. Priority has the largest event mass but a lower equal-cell mean (93.83%) than prefill (99.23%); fairness has only six positive cells. In the implemented first-applicable-stage taxonomy, resource feasibility erases priority and fairness disagreements, while the prefill filter erases prefill disagreements (Online Resource 1, Fig. S2). Zero residual classification ambiguity is conditional on this action representation and taxonomy. It is not proof of complete causal attribution or equality of future latency, ordering and state.

Unstratified-window and source-then-window sensitivities are retained in Online Resource 1, Sect. S9. The result remains conditional on the three selected pairs and the admission-control reference trajectory (Sect. 4.6); it does not extend the local estimand to future trajectories.

5.2.1 Extension to admission and KV guard

A separate preregistered extension evaluates earliest-deadline-first (EDF) scheduling with the Scorpio-labeled admission module and the Scorpio-labeled base with the KV-constrained donor's guard. On 21 unit-rate windows, 126 seed-level runs reduce to 42 pair–window cells because all three seeds give identical counts. Each family has 18 positive and three zero-denominator cells. Counting each window once, admission erases 103,159/103,421 disagreements (99.75%), whereas the KV guard erases 6,391/6,996 (91.35%). Thus high erasure extends beyond the original pairs, but its magnitude is not uniformly near 99%.

These descriptive event-weighted rates are not pooled with the original 56-window result: the pairs, reference policies and sampling frame differ. The extension's uncertainty concerns the mean positive-window rate, not the pooled

fraction (Online Resource 1, Sect. S12). Its modeled reserve guard inherits the compatibility limits in Sect. 7.2. The lower KV rate is a property of this pair and state distribution, not an isolated causal effect of module family.

5.3 Changed actions can affect requests without moving goodput

The full-trajectory prefill comparison answers the consequence question that a one-step erasure test cannot. In F1, completion time or SLO status changes for a mean fraction 0.588 of requests, with interval $[0.494, 0.684]$. The median absolute completion-time difference is 0.151 $[0.035, 0.301]$ in simulator time units, aggregated at the workload-instance level across the four prefill substitutions. Nevertheless, every one of those substitutions has $\Delta\text{ANWG} = 0$, and weighted mean tardiness and the number of SLO violations are unchanged. Figure 3 places those observations on the same diagnostic chain.

These findings are compatible: many completion times can move without crossing the binary deadline threshold in ANWG. A null aggregate effect therefore does not identify an inactive prefill module. Conversely, changed request timing does not establish a beneficial change: the reported fraction counts any change, not just improvement, and simulator time is not validated GPU latency. The observed zeros apply to the finite F1 design; no equivalence margin or population equivalence test is claimed.

5.4 Priority effects reach the aggregate endpoint under competition

In P1, the mean signed ANWG effect over the fixed priority inventory is $+0.0186$ $[0.0036, 0.0345]$, and 17/28 interventions meet the frozen FDR criterion. The P0–P1 contrasts in decision divergence and mean absolute ANWG effect both have permutation $p \approx 0.0005$. These findings concern variation across ten workload seeds per condition; they do not turn the 28 interventions on each seed into independent workload replications.

Effects are heterogeneous across base/donor combinations. The full priority forest, including null and harmful estimates, is in Online Resource 1, Fig. S3. A significant family-level pattern does not imply that every donor improves every base. The positive signed mean is also distinct from the mean absolute effect plotted in Fig. 2. It summarizes the tested inventory under competition, not a scheduler ranking or a portable deployment benefit.

6 Interactions and Robustness

The question is whether a module-level conclusion survives changes in its surrounding composition, workload and outcome measure. Four-arm contrasts address composition; held-out and trace-derived panels address workload transfer; the metric panel changes the endpoint on existing paired outputs. These designs are related diagnostic tests, not interchangeable replications.

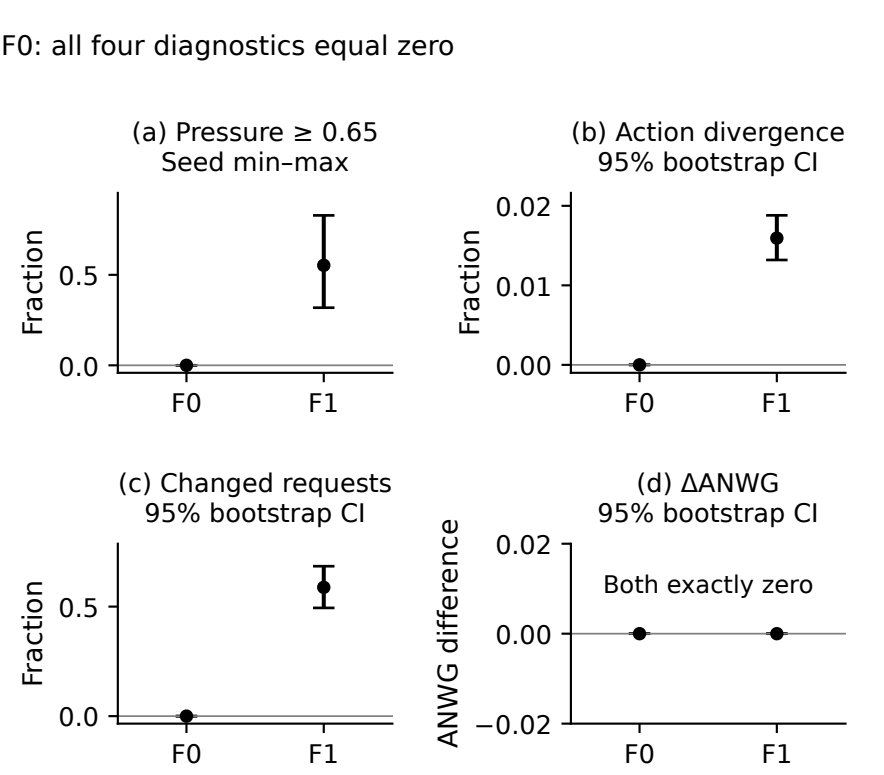


Fig. 3 Prefill multistage chain (prospective confirmatory; 10 workload instances, one per seed, per cell). F0 zeros remain visible as markers. (a) Pressure-active fraction (historical probe summary; seed min–max whiskers, not a CI). (b) Paired action divergence (95% CI). (c) Fraction of requests with changed completion timing or SLO status (95% CI). (d) ΔANWG (95% CI; exact zero under F1)

6.1 Modules can change one another's effects

The four-arm experiments expose non-additivity that isolated substitutions would miss. Standalone fairness is weak in the dedicated exploratory context (active mean $\Delta \approx -0.0119$, with interval including zero). In the controlled joint-priority condition, however, the priority–fairness interaction is $I = -0.0780$ $[-0.1170, -0.0355]$ and is FDR-significant. Fairness changes the context in which priority acts, so the isolated priority effect does not determine the joint effect.

The joint prefill condition illustrates why the sign of I requires care. Isolated prefill and fairness effects are harmful (-0.0289 and -0.0442 ANWG), but the joint degradation is less than their sum: $I = +0.0275$ $[0.0144, 0.0407]$. This is sub-additive degradation, not beneficial synergy. Figure 4 retains the isolated, additive-prediction, joint and interaction quantities; seed-level estimates are in Online Resource 1, Fig. S1. Each contrast uses ten paired work-

load instances. The weaker held-out and zero public-replay interaction results below limit any claim of a stable interaction magnitude.

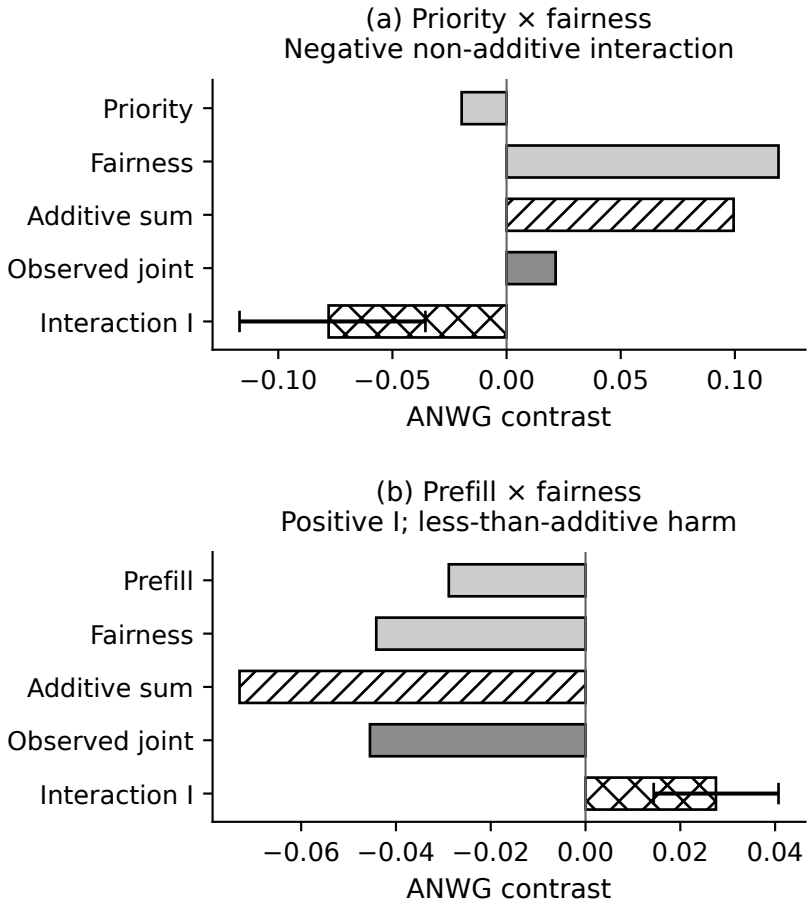


Fig. 4 Non-additive cross-module interactions (prospective confirmatory; 10 workload instances, one per seed, per condition). Bars show isolated module Δs, additive prediction $\Delta A+\Delta B$, observed joint Δ, and interaction I. Whiskers on I show its pointwise 95% CI. (a) Priority–fairness has a negative non-additive interaction ($I=-0.0780$). (b) Prefill×fairness exhibits sub-additive degradation ($I=+0.0275$)

6.2 Held-out workloads constrain transfer

Across 45 fresh synthetic workload instances, candidate-set opportunity is positively associated with priority divergence ($r = 0.85$). The low-versus-higher opportunity contrast has permutation $p = 0.0035$ and BH $q = 0.014$. Prefill

again shows action divergence with effectively zero ANWG change. These observations support the diagnostic distinctions outside the controlled instances, but not a universal effect size. In particular, the held-out priority–fairness interaction is -0.0018 $[-0.0126, 0.0071]$, with only 1/5 seed sign stability. The controlled interaction magnitude therefore does not transfer reliably. Exploratory held-out KV estimates are retained in Online Resource 1, Sect. S16, with the compatibility limitations in Sect. 7.2.

6.3 Production-derived replay has source-dependent relationships

The public replay design in Sect. 4.6 has 21 upstream windows and 63 dependent window–rate instances. The diagnostic correlations use one selected priority pair and one prefill pair, not all interventions in the replay inventory.

Prefill pressure is positively associated with action divergence ($r = 0.898$), and action divergence with request timing/SLO-status change ($r = 0.639$), while ANWG remains unchanged. Priority behaves differently: pooled structural opportunity is negatively associated with final divergence ($r = -0.241$). Six of seven source strata have positive within-source correlations, but Mooncake combines high structural opportunity with low realized divergence. The between-source correlation is -0.467. A pooled opportunity law would therefore conflate source composition with the behavior inside a source.

After source centering, the correlation is only 0.143, with corrected conditional bootstrap sensitivity range $[-0.015, 0.366]$. It includes zero. Resampling keeps all three rates of a source window together and recomputes the centering in each draw. The early/middle/late windows were deliberately selected and only three exist per source; these ranges do not establish population confidence coverage or a general within-source relationship. Full source/rate summaries and the other corrected ranges are in Online Resource 1, Sect. S10. Historical decision counts are absent from the stored diagnostic audits, preventing independent reconstruction of event-weighted denominators; the reanalysis uses the stored window ratios. The priority–fairness interaction is zero in all 63 observed instances. This is an observed design-specific null, not an equivalence result.

6.4 Actionable opportunity is a refinement, not a universal predictor

A prospective follow-up contains 140 new windows at three rates (420 window–rate cells). Primary inference uses the 140 unit-rate windows, with seed-level diagnostics averaged within each window. It tests whether actual top-choice disagreement is more informative than a nontrivial candidate set. Pooled actionable opportunity is directionally more correlated with divergence than structural opportunity (0.1285 versus 0.0740), but the difference 0.0545 has 95% interval $[-0.0197, 0.1839]$. The hypothesis of general superiority is not established. On prespecified Mooncake windows, the correlations are -0.3070

for structural and −0.0395 for actionable opportunity: disagreement reduces the mismatch without becoming a strong predictor.

The same campaign reports approximately 27.4% structural opportunity, 17.5% actionable opportunity, 2.4% final divergence and 82.5% actionable override. These are that campaign's stage summaries, not alternative estimates of the 99.3226% local same-state erasure fraction. Its windows, replay rates and trajectory diagnostics define a different population. Source-level results are in Online Resource 1, Table S2. The negative and inconclusive findings explain why opportunity should be measured along with downstream composition rather than used as a proxy for performance.

6.5 Endpoint choice changes numerical conclusions

The metric comparison contains 3,940 matched workload–intervention cells, each evaluated using ANWG and the four secondary endpoints defined in Sect. 3. After orienting all effects toward benefit, 3,028 cells (76.9%) are within 10^{-12} of zero on every endpoint; 655 (16.6%) change numerical detection status across endpoints; and 257 (6.5%) have opposing signs. Opposing signs take precedence over the detection category. These numerical classifications are neither significance tests nor equivalence tests; repeated interventions share workloads. Prefill is 92.2% near null on all endpoints, without implying request movement in every null cell. The finite metric set cannot establish invariance to all serving metrics. Online Resource 1, Sect. S26, gives the complete accounting and reconstruction checks.

The endpoints need not average the same outcomes: ANWG and weighted completion goodput normalize by all arrivals, whereas soft credit and tardiness use completion records (Sect. 3). Thus a change in who completes could change both the population and the endpoint. A post-hoc check of the saved panel finds weighted completion goodput exactly 1.0 in both arms of all 257 sign-sensitive cells, corroborated in their 36 source files. These reversals therefore coexist with unchanged reported completion coverage; they cannot be attributed to an observed loss of completion coverage. The aggregate records do not retain exact completion membership, so this check does not identify the request-level mechanism of each reversal. Different weighting and deadline penalties remain distinct even on a common population.

7 Validity of the Modeled Interventions

7.1 Recommendation agreement depends on context

The fidelity study compares a substituted component's recommendation with the modeled donor's recommendation on the same cloned state and candidate set. It evaluates 57 candidates on 63 window–rate configurations with three seed settings. Of 10,773 units, 8,121 contain reachable states; 15 context-dependent candidates are excluded. Table S9 in Online Resource 1 reports

the separate populations and denominators. Pooled agreement is 92.99%; the equal-weight positive-unit mean is 95.98%. Shared windows and seed variants are not independent workloads.

Admission, priority and the two tested KV guards agree on all evaluated reachable states. Fairness aging agrees on 28.87% of states (50.93% under equal positive-unit weighting), despite passing mutation and determinism checks. The modeled donor incorporates waiting time into its own priority key, whereas the substitution retains the base key and applies aging within that context. Different complete rankings therefore reveal a context mismatch, not a failure rate of a native fairness algorithm. The controlled interaction experiments use a separately specified promotion rule; their measured effects do not establish native-aging fidelity.

A separate prefill comparison tests four substitutions on 756 units at a common post-admission boundary. All 48,384 standard-view states agree, including 4,530 guard-active states. An overlapping active-state census also agrees on all 432,645 states; the views are not independent or additive. Donor and substitution invoke the same stateless modeled component, so agreement checks the boundary and state handling rather than independent implementation accuracy or native chunked-prefill fidelity. No unseen-state guarantee follows. Online Resource 1, Sect. S13, gives the exclusions, counts and implementation distinctions.

7.2 KV compatibility still limits native interpretation

The historical audit contains 72 transplant records and five slot-inventory records. Ten transplants are interface-flagged. Pinned-code checks show that five native-Scorpio interpretations are invalid and five reserve-donor interventions remain inconclusive. The Scorpio-labeled guard routes to a shared reserve-placement component, not the native Scorpio decode filter. The dynamic-fidelity campaign excludes all five flagged reserve candidates; its agreement on two other KV candidates does not validate those omissions.

The corrected KV figure retains six declared-slot pairs exploratorily; excluding flagged pairs leaves one unflagged pair. The mixed thirteen-row history includes seven contextual non-KV changes. Separate population estimates and seed-stream intervals remain in Online Resource 1, Sect. S16. Neither an unflagged status nor exact numerical recovery certifies native semantics. The original three-pair erasure result is unaffected by this KV exclusion; the new extension is interpreted only as modeled reserve behavior.

7.3 Real-engine execution is not verified paired confirmation

Initial real-vLLM confirmatory attempts failed infrastructure integrity and produced no valid primary estimate. Later prefill and native-priority retry campaigns completed 30 and 20 arms, respectively, and retained positive numerical analyses. The later prefill analysis required a repair to recognize its

frozen seed set; that repair reuses the same outputs and is not another execution.

The preserved pairing audit reports different realized input-token-length vectors *and multisets* in all 15 prefill pairs and all 10 priority-retry pairs. Equal trace-file hashes and client seed flags therefore do not establish identical served inputs. The positive contrasts remain recorded observations, but they cannot be promoted here to the intended identical-input paired treatment estimates. The process-dependent hashing in the inspected trace loader is a plausible explanation, not a proven historical cause. Online Resource 1, Sect. S19 gives the complete history and pairing audit. Strict real-engine paired confirmation remains unestablished.

7.4 Scope of inference

The experiments diagnose effects inside reconstructed compositions on selected workloads. Public traces supply arrivals and request shapes; priorities, SLOs and treatment outcomes are constructed. Local erasure conditions on the base state distribution and admitted-set representation. Full-trajectory effects include subsequent state changes. Neither is production causal identification. The campaigns are not a representative sample of module combinations. The comparator's unresolved utility questions are specified in Sect. 4.2; historical execution and public reconstruction boundaries are distinguished in Sect. 8. These qualifications limit transfer without erasing the measured recommendation, action and outcome differences.

8 Reproducibility and Evidence Availability

Public release v0.2.0 (linked in Code availability) provides eligible project-generated observations, analysis scripts, tests, statistical-unit definitions, provenance and SHA-256 checksums. It extends earlier synthetic substitution data [21] and the v0.1.0 outcome package with the equal-information pilot, erasure extension, fidelity counters, held-out observations, prospective source correlations and qualified later-engine summaries. The public repository is `https://github.com/SoroushVahidi/llm-scheduler-module-counterfactuals`.

The frozen release matrix supports saved-output numerical reconstruction for 36/48 enumerated principal claim families (75.00%), versus 14/48 in v0.1.0. Each family has equal weight; partially supported families, frozen answers and audit assertions receive no credit. This is neither a rate of independent experimental validation nor original-execution reproduction. The present correction to the trajectory-ambiguity denominator does not retroactively change that frozen release score. All 28 portable tests and the documented analysis commands run without private inputs; source identifiers and hashes do not promise access to private execution histories.

The package excludes the full historical simulator/serving environment, raw or reconstructive third-party traces and exact real-engine input vectors. Some original probe records, prediction vectors and execution attestations remain unavailable; native fidelity is not established by reconstructing saved counters. The pairing limitation is evaluated in Sect. 7.3. Online Resource 1, Sect. S14 maps the release files to these evidence boundaries. Exploratory prediction results remain frozen scores with limited test support (Sect. S18), not a reconstructed predictive-generalization result.

9 Discussion and Conclusion

The experimental specification makes the interpretation of a module substitution explicit: same-state comparisons test recommendation survival, while paired complete trajectories supply request and aggregate consequences. The measured erasure and request movement under aggregate nulls show why these observations should accompany a performance contrast. For an engineer interpreting a null, checking opportunity and downstream action changes distinguishes inactivity from filtering or an insensitive endpoint; request movement alone does not establish benefit.

Composition interactions require joint contrasts, and workload dependence and endpoint sensitivity require stating where and how an effect was measured. They do not support a portable module ranking. Equally instrumented ablation produces identical diagnoses in all seven constructed cases: the contribution is the declared diagnostic contract and scoped evidence, not a new causal estimator or demonstrated diagnostic advantage. Donor-context mismatches and failed real-engine pairing leave native fidelity and production transfer open.

Online Resource 1. Supplementary Information: extended intervention results, exploratory KV analyses, source/rate sensitivities, predictive baselines, matched-ablation contracts, fidelity checks, real-engine pairing diagnostics, and reproducibility details.

Statements and Declarations

Funding. The author received no specific funding for this research.

Competing interests. The author declares no competing interests. Complimentary Secure ShellFish access is disclosed as in-kind software support in the title-page acknowledgments.

Author contributions. Soroush Vahidi: conceptualization, methodology, software, data curation, formal analysis, visualization, writing—original draft, and writing—review and editing.

Code availability. MIT-licensed analysis code and tests are available in public release v0.2.0:
`https://github.com/SoroushVahidi/llm-scheduler-module-counterfactuals/releases/tag/v0.2.0`.

Data availability. Eligible project-generated outcomes and diagnostics are CC BY 4.0 in the same release, with upstream attribution. Reconstruction scope and exclusions are specified in Sect. 8; the versioned README gives the commands:
`https://github.com/SoroushVahidi/llm-scheduler-module-counterfactuals/blob/v0.2.0/README.md`. No license for third-party raw traces is implied.

Ethics and consent. Ethics approval and informed consent are not applicable: no human subjects participated and no human-subject research data were used. The research concerns scheduling experiments, simulated outcomes and third-party workload traces. This is a study-scope statement, not an institutional approval or formally issued exemption.

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. Agrawal A, Kedia N, Panwar A, et al (2024) Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve. In: 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24). USENIX Association, Santa Clara, CA, pp 117–134, URL `https://www.usenix.org/conference/osdi24/presentation/agrawal`
3. 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`
4. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B (Methodological) 57(1):289–300. `https://doi.org/10.1111/j.2517-6161.1995.tb02031.x`
5. 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`

6. 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
7. Efron B (1979) Bootstrap methods: Another look at the jackknife. The Annals of Statistics 7(1):1–26. https://doi.org/10.1214/aos/1176344552
8. 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
9. 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
10. Kwon W, Li Z, Zhuang S, et al (2023) Efficient memory management for large language model serving with PagedAttention. In: Proceedings of the 29th Symposium on Operating Systems Principles. ACM, SOSP '23, pp 611–626, https://doi.org/10.1145/3600006.3613165, URL https://dl.acm.org/doi/10.1145/3600006.3613165
11. Law AM (2015) Simulation Modeling and Analysis, 5th edn. McGraw-Hill Education
12. 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
13. 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
14. 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
15. 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
16. 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
17. 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
18. 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
19. Stojkovic J, Zhang C, Goiri Í, et al (2025) DynamoLLM: Designing LLM inference clusters for performance and energy efficiency. In: 2025 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, pp 1348–1362, https://doi.org/10.1109/HPCA61900.2025.00102
20. 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
21. Vahidi S (2026) Module intervention credit: Counterfactual single-module substitution data for LLM-serving scheduling policies. Hugging Face dataset, version 1, URL https://huggingface.co/datasets/SoroushVahidi/module-intervention-credit, revision 84430648ed5d564b29b4f9465efb27ad29921cf8. Historical synthetic module-substitution outcomes
22. Wang Y, Chen Y, Li Z, et al (2025) BurstGPT: A real-world workload dataset to optimize LLM serving systems. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2. ACM, KDD '25, pp 5831–5841, https://doi.org/10.1145/3711896.3737413
23. Yu GI, Jeong JS, Kim GW, et al (2022) Orca: A distributed serving system for Transformer-Based generative models. In: 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22). USENIX Association, Carlsbad, CA, pp 521–538, URL https://www.usenix.org/conference/osdi22/presentation/yu
24. 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

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ModuleCounterfactualsforAttributingLLMServingSchedulerEffectsOnlineResource1.pdf