FROM PRESSURE TO PRIORITY: CORRECTABLE PARTICLE TRANSPORT FOR DYNAMIC PARTITIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Distributed systems expose two views of contention that age at different rates: an occasional membership snapshot records which workloads shared a resource, while fresher local telemetry reports only how crowded each workload's own group appears. The latter behaves like a noisy observation of latent block cardinality, not like an identity-bearing edge measurement. A bootstrap partition filter ignores that distinction and spends its particles on split–merge transitions that fresh telemetry has already contradicted; replacing the belief by the most plausible single partition discards the ambiguity that same-cardinality groups necessarily carry. We introduce Pressure-Residual Partition Transport (PRPT), a full-support proposal for sequential inference over dynamic set partitions. PRPT factorizes a pressure-conditioned move into an event, a target cardinality, and a product-weighted fixed-cardinality subset. An elementary-symmetric-polynomial recursion samples splits and evaluates their proposal probabilities without enumerating exponentially many subsets, so exact importance correction remains available. The resulting kernel is permutation equivariant and costs $O(k|V| + k^2 + \sum_{C\in\Pi} |C|^2)$ for k blocks. With 32 particles under contaminated observations, PRPT lowers posterior total variation from 0.206 to 0.065 on an exactly enumerable benchmark, and removing correction raises it to 0.378. In delayed, lossy multi-step experiments, it improves a prespecified decision objective in all six test conditions. On 600 held-out Alibaba trace-derived episodes with mostly exact pressure, it reduces total variation by 0.042 and raises the effective-sample-size ratio by 0.211 relative to an equal-budget bootstrap filter; a separate 397-episode non-exact slice retains the ESS gain but exposes day-dependent posterior accuracy. These results delimit when cardinality telemetry improves finite-particle inference without pretending that it identifies membership.

1 INTRODUCTION

A scheduler rarely observes a shared computing system through one synchronized state. Placement and membership records are informative but expensive to aggregate; queue lengths, request counts, and utilization are cheap and arrive sooner. Consider a set of workloads whose latent contention groups form a partition. At time t the controller may hold yesterday's grouping $\widetilde{\Pi}_t$, while each workload reports a current estimate of the size of its own group. If four workloads all report size two, three pairings remain possible: the telemetry rules out most stale hypotheses, yet it never names partners. This separation between *fresh cardinality* and *stale membership* recurs in cluster control, distributed admission, and co-residency reasoning (Verma et al., 2015; Delimitrou & Kozyrakis, 2013; Ghodsi et al., 2011; Xiao et al., 2018; Gu et al., 2019; Qiao et al., 2021), and it sharpens the pressure-versus-stale-state intuition of DFD-CR (Hu & Ren, 2026): here the pressure observes a latent discrete structure instead of acting as a force or priority rule.

The latent state is combinatorial, and this is where existing machinery strains. A partition of n objects ranges over a Bell-number space, and one binary split of a block of size m already has $2^{m-1} - 1$ outcomes—32,768 for $m = 16$ and 2.15×10^9 for $m = 32$. Sequential Monte Carlo (SMC) is a natural fit because it represents posterior ambiguity and supports online decisions (Gordon et al., 1993; Doucet et al., 2001; Del Moral, 2004; Naesseth et al., 2019), but its value is set by the proposal,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and neither standard option survives this setting. A bootstrap filter proposes from the transition prior and consults the fresh pressure only after weighting, so a small particle budget is spent on cardinalities the observation has already ruled out (Snyder et al., 2008). An enumerated likelihood-guided proposal avoids that waste by scoring every successor, and in doing so recreates the exponential split support it was meant to navigate.

This paper develops a proposal operator around the semantics of the observation. For a particle Π, the residual between a node's reported cardinality and its cardinality under Π supplies evidence at three nested levels: whether the next event is a stay, merge, or split; what block size the event should produce; and which nodes form the resulting fixed-size subset. Pressure-Residual Partition Transport (PRPT) samples these choices hierarchically. The final subset is drawn from a product-weighted fixed-cardinality law through an elementary-symmetric-polynomial (ESP) dynamic program, so the sample arrives with its exact proposal probability q, and the pressure guidance changes variance rather than the filtering target. A defensive prior mixture gives every allowed transition positive probability.

The construction deliberately uses established ingredients—SMC, split–merge moves, robust residuals, defensive mixtures, and weighted subset sampling (Jain & Neal, 2004; Hughes et al., 2012; Owen & Zhou, 2000; Huber, 1964; Chen et al., 1994). The new object is their coupling under a node-wise own-block-cardinality observation: residual $\rightarrow$ event $\rightarrow$ target size $\rightarrow$ fixed-size membership subset, with non-enumerative sampling and tractable correction. This differs from adding new data points to a clustering model, from partitioning state dimensions to run a block filter, and from sequentially constructing a phylogeny (Ulker et al., 2010; Min et al., 2022; Moretti et al., 2021; Trojan et al., 2026).

Our contributions are:

- We formulate asynchronous cardinality sensing for dynamic latent partitions and show why cardinality narrows, but generally cannot resolve, membership ambiguity.
- We derive PRPT, a full-support, exactly correctable proposal whose binary-split sampler is polynomial rather than exponential in block size. We establish permutation equivariance and give explicit complexity bounds.
- We separate proposal quality from posterior correctness through exact small-state evaluation, correction ablations, multi-step decisions, and CPU scaling. On held-out contaminated cases, the factorized proposal improves total variation and effective sample size over equal-budget bootstrap SMC.
- We test the frozen method on five previously unseen days of the Alibaba GPU-cluster trace. The evaluation reports clean and non-exact telemetry strata separately, preserves a negative day, and uses day–episode hierarchical intervals rather than treating all episodes as independent.

2 Problem setting and opportunity

Dynamic partition. Let $V = \{1, \dots, n\}$ be a fixed node set and $\Pi_t = \{C_{t,1}, \dots, C_{t,k_t}\}$ a set partition, writing $C_\Pi(i)$ for the block containing i. The transition prior $p(\Pi_t \mid \Pi_{t-1})$ permits one local event: stay, merge any two blocks, or split one block into two nonempty blocks. This deliberately local model suits observation intervals that are short relative to churn; batched changes are discussed in Section 7.

Two observation channels arrive asynchronously. A delayed or lost control-plane snapshot reveals an earlier partition and re-anchors the particles when received. Between snapshots, each node emits

$$y_{i,t} = |C_{\Pi_t}(i)| + \epsilon_{i,t}, \tag{1}$$

where ϵ may be Gaussian or contaminated. We use "pressure" for y because local demand and request telemetry often scale with the number of contenders; the method only requires a cardinality-calibrated observation model. It does not infer arbitrary graph edges from scalar load.

Why a point estimate is insufficient. Cardinality is invariant to relabeling and can be highly informative. In an exact $n = 8$ enumeration with pairwise edit budget six, the stale prior admits

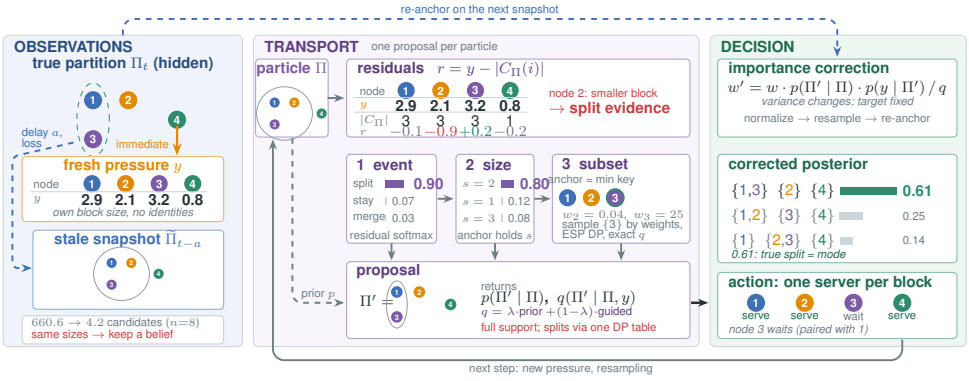


Figure 1: One PRPT transport step. *Left (observations):* a hidden partition is seen through two channels of different age—a delayed, lossy membership snapshot and fresh noisy node-wise block sizes; the worked example tracks node 3 splitting off $\{1,2,3\}$. *Middle (transport):* per particle, residuals between reported and implied block sizes score a three-level move (event, target size, fixed-size subset), every level mixed with the transition prior; the subset is sampled through an ESP recursion, never by listing $2^{|C|-1}-1$ splits, and the proposal returns its exact p and q. *Right (decision):* importance weights correct the guidance, the belief concentrates on the true split, and a convention action follows by posterior expected utility. Dashed lanes: snapshot re-anchoring (top) and the next step (bottom).

660.6 partitions on average; noiseless pressure leaves 4.2. Yet repeated block sizes prevent unique recovery: $\{\{1,2\},\{3,4\}\}$, $\{\{1,3\},\{2,4\}\}$, and $\{\{1,4\},\{2,3\}\}$ share one cardinality vector. The task is partial identification, not ordinary graph reconstruction. Bayesian partition models express exactly this kind of uncertainty (Kingman, 1978; Pitman, 2006; Wade & Ghahramani, 2018; Orbanz & Roy, 2015); constrained-size clustering supplies optimization primitives but no dynamic filtering rule (Puleo & Milenkovic, 2016; Paganin et al., 2021).

Decision interface. The filter estimates $p(\Pi_t \mid \widetilde{\Pi}_{1:t}, y_{1:t})$. A downstream convention action a_t is selected by posterior expected utility, $\arg\max_a \mathbb{E}[U(a, \Pi_t)]$. The experiments use mutually exclusive component service as a transparent probe: a wrong convention causes simultaneous service (a violation), whereas an overly cautious convention idles. This is not reinforcement learning; it is a belief-space decision used to test whether an inference improvement survives contact with an action.

3 Pressure-Residual Partition Transport

3.1 A hierarchical proposal

For a source particle Π define node and block residuals

$$r_i = y_i - |C_\Pi(i)|, \qquad R(C; s) = \sum_{i \in C} \rho(y_i - s)\,, \tag{2}$$

where ρ is squared or Huber loss (Huber, 1964). Low residual under the current block size supports a stay; two blocks whose nodes agree with the merged size support a merge; heterogeneous residuals inside one block support a split. Rather than scoring every successor partition, PRPT defines

$$q_g(\Pi' \mid \Pi, y) = q_E(e \mid \Pi, y)\, q_T(s \mid e, \Pi, y)\, q_S(A \mid s, e, \Pi, y), \tag{3}$$

where e is an event, s a target size, and A the anchor-side subset of a split. Stay and merge leave factors degenerate. Event and size logits are negative residual costs divided by a temperature τ.

For a split of C with anchor $a \in C$ and target size s, let $A \subset C$, $a \in A$, $|A| = s$. Node log-odds ℓ_i compare the residual cost of placing i on the two sides of sizes s and $|C| - s$. The conditional subset law is

$$q_S(A \mid s, C, y) = \frac{\prod_{i \in A \setminus \{a\}} \exp(\ell_i)}{e_{s-1}(\{\exp(\ell_j) : j \in C \setminus \{a\}\})}, \tag{4}$$

Algorithm 1 One corrected PRPT update

Given: weighted particles $\{(\Pi^{(m)}, w^{(m)}, \kappa^{(m)})\}_{m=1}^{M}$, pressure y, optional snapshot $\widetilde{\Pi}$
for $m = 1, \dots, M$ **do**
  if a snapshot arrived **then** re-anchor $\Pi^{(m)} \leftarrow \widetilde{\Pi}$, keeping keys $\kappa^{(m)}$
  compute node and block residuals from Equation (2)
  draw event e: with probability λ from the transition hierarchy, otherwise from residual scores
  if e is a split **then**
    draw target size s, orient the block by its smallest key, build the ESP table, sample A ▷ Equation (4); no enumeration
  end if
  return $\Pi'^{(m)}$, $p_m = p(\Pi'^{(m)} \mid \Pi^{(m)})$, and the complete mixture probability q_m
  $\widetilde{w}^{(m)} \leftarrow w^{(m)}\, p_m\, p(y \mid \Pi'^{(m)})/q_m$
end for
normalize $\widetilde{w}$; resample at the ESS threshold (Douc et al., 2005)
choose the action by posterior expected utility, $\arg\max_a \mathbb{E}[U(a, \Pi_t)]$

where e_d is the degree-d elementary symmetric polynomial. The recursion $E[j, d] = E[j-1, d] + w_j E[j-1, d-1]$ evaluates the normalizer and supports backward sampling in $O(|C|^2)$ time. Related recursions appear in unequal-probability sampling and fixed-cardinality point processes (Chen et al., 1994; Kulesza & Taskar, 2012; Kool et al., 2019); here one makes a pressure-conditioned partition transition both sampleable and exactly evaluable.

An unordered split needs a canonical orientation. Each node carries an exchangeable continuous key sampled once; the smallest-key node is the anchor. Keys move with nodes under relabeling and carry no semantic identity, which prevents counting both A and $C \setminus A$ while preserving distributional symmetry (Bloem-Reddy & Teh, 2020; Zaheer et al., 2017; Murphy et al., 2019).

3.2 Support and importance correction

Guidance alone can assign negligible mass to a valid but surprising event. We therefore mix the guided hierarchy with the transition prior:

$$q_\lambda(\Pi' \mid \Pi, y) = \lambda p(\Pi' \mid \Pi) + (1-\lambda) q_g(\Pi' \mid \Pi, y), \quad 0 < \lambda < 1. \tag{5}$$

This is a fixed defensive mixture, not an adaptive-mixture contribution (Hesterberg, 1995; Owen & Zhou, 2000; Branchini & Elvira, 2021). Given ancestor weight w, the incremental unnormalized weight is

$$w' = w \frac{p(\Pi' \mid \Pi) p(y \mid \Pi')}{q_\lambda(\Pi' \mid \Pi, y)}. \tag{6}$$

We normalize, resample when the effective sample size demands it (Douc et al., 2005), and re-anchor all particles when a new discrete snapshot arrives. The observation model in Equation (6) may differ from the residual family used to propose; this separation permits robust guidance without silently changing the posterior target.

Inference rather than hand-coded reconstruction. The residual score is fixed, but the output is a distribution over a combinatorial latent state and every decision consumes that belief. In machine-learning terms, PRPT is a structured amortization of a local Bayesian update: the observation maps directly to a proposal family whose support and density are known, which separates it from a deterministic heuristic that maps telemetry to one reconstructed graph. It also identifies the disciplined extension: replace the residual logits by a learned permutation-equivariant scorer while preserving the normalized event/size/subset factors and Equation (6). Training such a scorer could use filtering variational objectives or differentiable resampling (Maddison et al., 2017; Naesseth et al., 2018; Corenflos et al., 2021); neither training nor its benefits are claimed here.

3.3 Properties

Proposition 1 (support and validity). If the transition prior has positive probability on every allowed one-step neighbor and $\lambda > 0$, then $q_\lambda(\Pi' \mid \Pi, y) > 0$ wherever $p(\Pi' \mid \Pi) > 0$. Equation (6) is

Table 1: Evaluation matrix. Each benchmark answers one question: (Q1) finite-particle posterior quality against an enumerable target; (Q2) necessity of factorization and correction; (Q3) multi-step decisions under stale communication; (Q4) held-out trace-derived observations and scale. "Exact" means the target posterior is enumerated only for scoring.

Q	Benchmark	Scale and split	Principal comparisons	Metrics
Q1	Exact one-step synthetic	$n = 8$; Gaussian and contaminated pressure; 100 validation, 200 test cases per regime	bootstrap; enumerated Gaussian guide; exact proposal; PRPT without correction	TV, ESS, coverage, regret, invalid actions
Q2	Factorized gate	500 new $n = 8$ transitions; 32 particles	bootstrap; misspecified enumerated guide; correction ablations in §5	TV, ESS, coverage
Q3	Delayed/lossy trajectories	4 nodes; 6 held-out conditions $\times$ 100 paired episodes; 80 steps	bootstrap; no correction; robust-pressure rule; stale persistence; current-view oracle	objective, violations, productivity, completions
	Random topology	$n \in \{8, 12\}$; 6 unseen churn/channel conditions $\times$ 100	adds an equally tuned L2 guide and age backoff	paired objective/violation intervals
	Independent domains	$n \in \{4, 8, 16\}$; 50 episodes per size	bootstrap; pressure rule; stale persistence; oracle	objective vs. oracle, violations
Q4	Alibaba ASI v2026 trace	2 mostly-exact days / 600 episodes; 3 non-exact days / 397	bootstrap; L2/Huber/mixture guides; no correction; request rule; stale; exact posterior	TV, MAP, coverage, ESS, utility, violations
	CPU cost	blocks up to $n = 64$; 500 samples per size	explicit support construction at $n = 16$	median/p95 time, support size

therefore finite, and standard SMC consistency applies for bounded test functions. The result follows from $q_\lambda \geq \lambda p$ and the ordinary importance identity (Del Moral, 2004; Chopin & Papaspiliopoulos, 2020).

Proposition 2 (permutation equivariance). Let a permutation σ act jointly on node labels, observations, partitions, and persistent keys. Then $q_\lambda(\sigma\Pi' \mid \sigma\Pi, \sigma y) = q_\lambda(\Pi' \mid \Pi, y)$. Block residuals are sums, event scores depend only on blocks, ESPs are symmetric polynomials, and the anchor key follows its node. The particle law is therefore equivariant even though a sampled realization may pick one of several symmetric memberships.

Proposition 3 (complexity). For k blocks, computing event evidence and sampling one move costs

$$O\left(k|V| + k^2 + \sum_{C \in \Pi} |C|^2\right), \tag{7}$$

with quadratic memory only for the largest-block ESP table. Directly listing all binary splits costs $\sum_C (2^{|C|-1} - 1)$. The claim concerns one proposal, not end-to-end filtering at a fixed particle count. Proofs and probability-normalization checks are in Appendix A.

4 Experimental design

The experiments answer four questions, mapped to benchmarks in Table 1: (Q1) does pressure guidance improve a finite-particle posterior when exact evaluation is possible; (Q2) are factorization and correction individually necessary, tested by ablations inside the Q1 and Q4 benchmarks; (Q3) does posterior quality improve multi-step decisions under stale communication; and (Q4) does the frozen method survive trace-derived observations and larger support? Hyperparameters are selected on separate validation seeds or days. Unless stated otherwise, PRPT uses 32 particles and a 50–50 mixture of prior and L2-guided particles, and bootstrap SMC receives the same budget. Exact-posterior methods are oracles, not deployable competitors.

Synthetic exact posterior. Random $n = 8$ partitions undergo one stay, merge, or split. Pressure is Gaussian or a 90/10 contaminated Gaussian. The one-step support is enumerated only to obtain

Table 2: Contaminated-pressure exact test, 32 particles (200 cases). ESS for no-correction is not a valid posterior ESS because proposal samples are treated as equally weighted.

Method	TV↓	ESS↑	Cover.↑	Regret↓
Bootstrap	0.2062	0.135	84.5%	0.680
Gaussian guided	0.1804	0.934	83.5%	0.495
Exact proposal	0.0619	1.000	98.5%	0.340
PRPT	**0.0645**	0.621	**99.0%**	**0.275**
PRPT, no correction	0.3778	1.000*	99.0%	0.535

ground truth. Baselines are the transition-prior bootstrap filter, an enumerated Gaussian-likelihood proposal, an exact-posterior proposal, and PRPT without p/q correction. Metrics are posterior total variation (TV), ESS divided by particle count, true-state particle coverage, decision regret, and invalid-action rate. Parameters ($\tau = 0.5$, exploration 0.2, Huber threshold 0.5) are fixed using 100 contaminated validation cases; each test regime has 200 new cases. A second gate evaluates factorized PRPT on 500 further cases.

Delayed multi-step decisions. Four nodes follow split–merge trajectories for 80 steps. Six held-out conditions vary snapshot delay 1–4, message loss 0–0.3, and pressure noise 0–1. Each contains 100 paired episodes with shared environment randomness. We compare bootstrap, no-correction, a frozen robust-pressure point rule, stale persistence, and a current-view oracle. The prespecified objective is productive component-steps $-2\times$ violations $+0.25\times$ completions. Independent-domain scaling composes the same process over 4, 8, and 16 nodes.

To avoid testing only a periodic four-node path, a second protocol freezes the implementation before generating six unseen random-topology conditions: $n \in \{8, 12\}$, event probability $0.08/0.18/0.30$, delay $1/2/4$, loss $0.05/0.15/0.30$, and noise $0.25/0.5/0.75$. Each contains 100 independently seeded 80-step trajectories; merges may cross all historical blocks. Bootstrap, a squared-residual (L2) proposal, a pressure rule, and age backoff receive equal validation-only tuning. Test conditions are neither removed nor reused for selection.

Alibaba trace-derived benchmark. We construct server co-residency partitions from the public Alibaba GPU-cluster trace (Alibaba Cloud, 2026). This is a server-level proxy, not GPU-device placement or a replay of production job completion time. Day 30 calibrates observation mappings and day 60 selects the frozen variant. Test days 170/180 yield 600 one-arrival episodes where request pressure is mostly an exact cardinality proxy. Before inspecting new days, we freeze a non-exact protocol; days 160/175/184 provide 397 episodes where at least one node's pressure differs from its block size. Alongside bootstrap and the synthetic baselines, we include Huber and contaminated-likelihood defensive proposals, stale persistence, and a deterministic request-residual rule. Intervals resample days first and episodes within days second. The small number of days makes these sensitivity intervals, not population-wide guarantees.

Implementation checks. The codebase has 109 tests covering partition validity, support normalization, sampler–evaluator agreement, underflow, relabeling equivariance, snapshot re-anchoring, and dataset hashes. CPU timings use Ubuntu 22.04 LTS, Python 3.13.15, and NumPy 2.5.3; each proposal benchmark has 20 warmups and 500 timed samples.

5 Results

5.1 Exact posterior and component ablations

Table 2 and Figure 2a isolate the mechanism. Under contamination, PRPT nearly matches the exact-posterior proposal in TV and improves on the misspecified Gaussian guide despite its lower ESS—a caution against reading ESS alone as accuracy. Without correction, coverage stays high because the proposal visits the truth, yet TV rises sixfold: directed transport helps only when its probability is accounted for. Under correctly specified Gaussian noise, enumerated Gaussian guidance

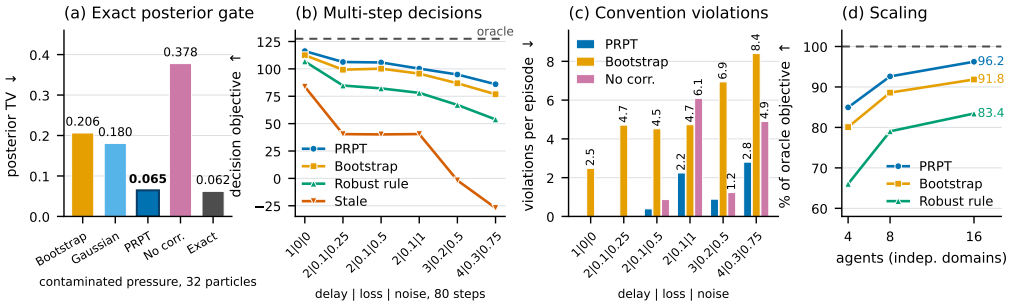


Figure 2: Synthetic evidence. (a) Under contaminated pressure, corrected guidance approaches the exact posterior at 32 particles; the uncorrected proposal is biased. (b) Across all six delayed/lossy conditions, PRPT holds the highest non-oracle objective; the dashed line is the current-view oracle. (c) The same conditions show fewer convention violations than equal-budget bootstrap and no-correction variants. (d) On explicitly independent local domains, PRPT stays closest to the oracle objective as the agent count grows. Error intervals and full tables are in the appendix.

and PRPT are essentially tied (TV 0.0520 and 0.0518 at 32 particles); we do not claim dominance over a correctly specified, enumerable optimal guide.

Factorization retains the gain on 500 new contaminated cases: TV is 0.0885 for PRPT, 0.1120 for bootstrap, and 0.1631 for the misspecified enumerated Gaussian guide. ESS ratios are 0.428, 0.156, and 0.952, respectively; true-state coverage is 92.8%, 92.0%, and 84.8%. The non-enumerative operator thus lowers TV by 21.0% relative to bootstrap while raising ESS by $2.75\times$.

5.2 Dynamic decisions and scale

Across the six multi-step conditions (Figure 2b,c), PRPT improves the objective over bootstrap in every condition, with a mean gain of 6.36 per episode and 4.24 fewer violations. Its advantage over no-correction is 3.13 objective and 1.13 violations. The hardest condition (delay 4, loss 0.3, noise 0.75) gives objectives 85.99 for PRPT, 76.84 for bootstrap, 53.85 for the pressure rule, and -27.2 for stale persistence; the current-view oracle is 127.5. The gap to the oracle is expected: cardinality does not reveal same-size membership, and snapshots can be absent.

The independent-domain test (Figure 2d) asks a narrower scaling question than arbitrary global partitions. With 16 nodes, PRPT reaches 96.2% of oracle objective and incurs 0.88 violations per episode, compared with 91.8% and 21.38 for bootstrap. Independence is supplied to every method; the experiment shows that local posteriors compose, not that PRPT discovers a domain decomposition.

Random-topology blind test. The advantage is not confined to the periodic trajectory. Relative to equal-budget bootstrap, PRPT raises mean objective in all six random-topology conditions: gains range from 3.84 (95% paired interval $[2.88, 4.80]$) at $n = 8$ with low churn to 19.90 $[14.95, 24.67]$ at $n = 12$ with high churn. Violations fall by 0.75–6.01 at $n = 8$ and 1.33–5.13 at $n = 12$. Against an equally tuned informed proposal, Huber-PRPT and L2 guidance tie in four conditions; L2 is better in the low-churn $n = 12$ case by 0.71 and in the hardest $n = 12$ case by 5.82. PRPT still beats tuned robust-pressure objective in all conditions, but in the hardest case the rule has 2.95 fewer violations. The evidence therefore supports corrected pressure-informed transport, not universal superiority of the Huber residual family. Across the six conditions PRPT reaches 71.7%–97.6% of oracle objective, leaving a visible gap where delay, churn, and block size grow together.

5.3 Held-out trace-derived partitions

The mostly-exact stratum supports the finite-particle claim across posterior and decision metrics (Table 3). The non-exact stratum is deliberately less tidy. ESS and violation intervals remain positive, while TV and MAP intervals cross zero and coverage decreases on two of three days. Day 175 is negative for TV, MAP, and coverage, yet positive for utility and violation reduction. Normalization,

Table 3: Frozen PRPT minus equal-budget bootstrap on Alibaba test days. Positive is better. Intervals use day–episode hierarchical bootstrap.

Metric	Mostly exact (2d/600)	Non-exact (3d/397)
TV reduction	$0.0420\ [0.0288, 0.0557]$	$0.0345\ [-0.0238, 0.0761]$
MAP gain	$0.0450\ [0.0200, 0.0733]$	$0.0096\ [-0.0628, 0.0626]$
Coverage gain	$0.0333\ [0.0167, 0.0533]$	$-0.0403\ [-0.1208, 0.0166]$
ESS-ratio gain	$0.2110\ [0.2020, 0.2201]$	$0.0494\ [0.0147, 0.0756]$
Utility gain	$0.0617\ [0.0233, 0.1067]$	$0.1134\ [-0.0300, 0.2642]$
Violation reduction	$0.0167\ [0.0033, 0.0300]$	$0.0455\ [0.0002, 0.0944]$

Table 4: Per-day paired mean changes versus bootstrap. The five-day aggregate is descriptive because the first two and last three days use different pressure-slice criteria.

Day (N)	TV↓	ESS↑	Utility↑	Viol.↓
170 (300)	0.0413	0.2111	0.0700	0.0200
180 (300)	0.0427	0.2109	0.0533	0.0133
160 (128)	0.0563	0.0683	0.0547	0.0234
175 (69)	-0.0166	0.0118	0.1304	0.0580
184 (200)	0.0638	0.0680	0.1550	0.0550
equal-day mean	0.0375	0.1140	0.0927	0.0339
95% hierarchical	[0.0013,0.0635]	[0.0422,0.1844]	[0.0051,0.1947]	[0.0064,0.0698]

support, underflow, partition validity, hashes, and repeated particle seeds do not identify a software fault. The likely cause is support loss when the fixed L2 residual family meets both a large candidate set and shifted observation semantics. Reporting this day matters: the method is a correctable proposal, not a guarantee that every guide improves every finite-particle approximation.

The correction ablation is decisive on real data. On days 170/180, no-correction TV is $0.5700/0.5757$ and MAP accuracy $0.7033/0.6800$; corrected PRPT obtains TV $0.1063/0.1214$ and MAP $0.9533/0.9467$. On the three non-exact days, no-correction TV lies between 0.752 and 0.799. The gains therefore do not come from relabeling guided samples as posterior draws.

The per-day view in Table 4 separates three statements that a pooled score would conflate. First, guidance increases ESS and improves the decision metrics on every observed day. Second, distributional accuracy is not uniformly improved: day 175 is a counterexample, and coverage is positive on only two of five days. Third, higher ESS does not certify a better posterior when the guide concentrates on the wrong region—the same lesson the misspecified Gaussian proposal teaches in Table 2. Proposal evaluation should therefore pair a weight-degeneracy measure with exact or task-relevant accuracy whenever a tractable oracle exists.

Where the evidence supports the claim. The one-step oracle test identifies posterior error; the factorized gate checks that the same operator does not require enumeration; correction ablations distinguish guided sampling from biased inference; multi-step and random-topology tests show that the difference reaches actions; and the trace study tests the observation-to-proposal mapping outside the procedural generator. No single experiment establishes all five links. Taken together, they support the bounded statement that structured cardinality guidance improves finite-particle inference in the tested regimes while preserving a valid target. They do not support universal finite-M dominance, exact recovery of repeated-size partitions, or production scheduling benefits.

5.4 Cost

At 16 nodes, factorized sampling is about $600\times$ faster than constructing the split support (Table 5). End-to-end 32-particle decision latency is $2.77/4.40$, $4.17/8.48$, and $5.86/16.21$ ms median/p95 for 8, 12, and 16 nodes. Bootstrap is cheaper (2.54, 3.36, and 4.68 ms median) because it computes no residual scores. These Python/WSL timings demonstrate feasibility at the tested scales; they are not production latency claims.

Table 5: CPU time for one factorized proposal. Split support counts possible binary splits; enumeration is measured only where feasible.

n	max $\lvert C\rvert$	split support	median / p95	enum. median
16	16	32,768	0.211/0.323 ms	127.1 ms
32	32	2.15×10^9	0.833/1.40 ms	–
64	32	4.29×10^9	1.50/2.31 ms	–

6 Related work

Particle inference. Bootstrap and auxiliary filters established the proposal–weighting view used here (Gordon et al., 1993; Pitt & Shephard, 1999; Liu & Chen, 1998). Subsequent work learns proposals, optimizes their allocation, differentiates through resampling, or treats SMC as a variational family (Gu et al., 2015; Naesseth et al., 2018; Maddison et al., 2017; Le et al., 2018; Corenflos et al., 2021; Branchini & Elvira, 2021). Those are general adaptation strategies; PRPT instead derives a normalized proposal from the specifics of a combinatorial observation.

Partitions and split–merge inference. Data-informed moves recur in partition SMC, split–merge MCMC, particle Gibbs, and community detection (Ulker et al., 2010; Jain & Neal, 2004; Hughes et al., 2012; Bouchard-Côté et al., 2017; Peixoto, 2020). Combinatorial SMC also builds phylogenies or decomposes growing clustering problems (Moretti et al., 2021; Trojan et al., 2026), while dynamic block models infer communities from edges (Xu & Hero, 2014; Matias & Miele, 2017). None of these combines stale membership with fresher own-block cardinality, and none provides our event–size–fixed-subset transition. Our claim is not the first partition SMC or informed split–merge method.

Learning and systems context. Exchangeability explains why relabeling nodes should not change a partition law (Orbanz & Roy, 2015; Bloem-Reddy & Teh, 2020; Zaheer et al., 2017); persistent keys orient our samples without fixed semantic IDs. Cluster systems study placement, fairness, preemption, and goodput (Verma et al., 2015; Delimitrou & Kozyrakis, 2013; Ghodsi et al., 2011; Xiao et al., 2018; Gu et al., 2019; Qiao et al., 2021; Mao et al., 2019), and their traces motivate our observation model (Reiss et al., 2012). PRPT estimates latent contention for a downstream rule; it is not a scheduler, and our trace test does not establish JCT or SLA gains.

7 Limitations and discussion

The transition model allows one binary event per interval. High churn may require composing multiple moves, shortening the interval, or enlarging the event vocabulary. The ESP sampler removes split enumeration but remains quadratic in the largest block. Pressure must be calibrated to own-block cardinality; arbitrary utilization signals may violate Equation (1). Repeated sizes remain intrinsically ambiguous, and no proposal can recover information the observation never carried.

The real-data evidence covers five days and server co-residency, not device-level interference or all future shifts. The non-exact slice shows that a robust proposal can still lose support. Richer likelihoods or learned equivariant scores may help, but a successor method must preserve a normalized subset law and transfer across size, churn, and noise—not merely attach a graph network to residuals.

The contribution is a probabilistic inference kernel over combinatorial structures, its symmetry, and its finite-particle behavior. The convention test is a decision diagnostic, not an RL benchmark; this boundary keeps the claim commensurate with the evidence.

8 Conclusion

PRPT turns fresh cardinality pressure into corrected proposals over events, sizes, and subsets without treating pressure as membership. Exact, dynamic, and trace-derived tests locate the finite-particle gain and its failure under observation shift. Asynchronous side information is most useful when its invariances and its ambiguities shape the proposal rather than a point estimate.

REPRODUCIBILITY STATEMENT

Appendix B specifies data construction, splits, metrics, and hyperparameters; Appendix A supplies proof details. The supplementary code regenerates all synthetic data and paper artifacts, downloads documented public trace shards, verifies source/data hashes, and runs 109 tests. Raw third-party trace files are not redistributed.

AI USE STATEMENT

In this work, we used generative AI tools for content refinement and polish. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Alibaba Cloud. Alibaba cluster trace program: Gpu cluster trace v2026. https://github.com/alibaba/clusterdata/tree/master/cluster-trace-gpu-v2026, 2026. Accessed 2026-09-25.

Benjamin Bloem-Reddy and Yee Whye Teh. Probabilistic symmetries and invariant neural networks. *Journal of Machine Learning Research*, 21:1–61, 2020. URL https://jmlr.org/papers/v21/19-322.html.

Alexandre Bouchard-Côté, Arnaud Doucet, and Andrew Roth. Particle gibbs split-merge sampling for bayesian inference in mixture models. *Journal of Machine Learning Research*, 18(28):1–39, 2017. URL https://jmlr.org/papers/v18/15-397.html.

Nicola Branchini and Víctor Elvira. Optimized auxiliary particle filters: adapting mixture proposals via convex optimization. In *UAI*, volume 161 of *PMLR*, pp. 1289–1299, 2021. URL https://proceedings.mlr.press/v161/branchini21a.html.

X.-H. Chen, Arthur P. Dempster, and Jun S. Liu. Weighted finite population sampling to maximize entropy. *Biometrika*, 81(3):457–469, 1994. doi: 10.1093/biomet/81.3.457.

Nicolas Chopin and Omiros Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer, 2020. doi: 10.1007/978-3-030-47845-2.

Adrien Corenflos, James Thornton, George Deligiannidis, and Arnaud Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *ICML*, volume 139 of *PMLR*, pp. 2100–2111, 2021. URL https://proceedings.mlr.press/v139/corenflos21a.html.

Pierre Del Moral. *Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, 2004. doi: 10.1007/978-1-4684-9393-1.

Christina Delimitrou and Christos Kozyrakis. Paragon: Qos-aware scheduling for heterogeneous datacenters. In *ASPLOS*, pp. 77–88, 2013. doi: 10.1145/2451116.2451125.

Randal Douc, Olivier Cappé, and Eric Moulines. Comparison of resampling schemes for particle filtering. *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pp. 64–69, 2005. doi: 10.1109/ISPA.2005.195385.

Arnaud Doucet, Nando de Freitas, and Neil Gordon (eds.). *Sequential Monte Carlo Methods in Practice*. Springer, 2001. doi: 10.1007/978-1-4757-3437-9.

Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: fair allocation of multiple resource types. In *NSDI*, pp. 323–336, 2011.

Neil J. Gordon, David J. Salmond, and Adrian F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F*, 140(2):107–113, 1993. doi: 10.1049/ip-f-2.1993.0015.

Juncheng Gu, Mosharaf Chowdhury, Kang G. Shin, Yibo Zhu, Myeongjae Jeon, Junjie Qian, Hongqiang Liu, and Chuanxiong Guo. Tiresias: a gpu cluster manager for distributed deep learning. In *NSDI*, pp. 485–500, 2019. URL https://www.usenix.org/conference/nsdi19/presentation/gu.

Shixiang Gu, Zoubin Ghahramani, and Richard E. Turner. Neural adaptive sequential monte carlo. In *NeurIPS*, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/99adff456950dd9629a5260c4de21858-Abstract.html.

Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194, 1995. doi: 10.1080/00401706.1995.10484303.

Zhenqing Hu and Bin Ren. DFD-CR: Decentralized fluid dynamics for congestion resolution at multi-agent bottlenecks. Poster presented at the IEEE High Performance Extreme Computing Conference (HPEC), 2026.

Peter J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1): 73–101, 1964. doi: 10.1214/aoms/1177703732.

Michael C. Hughes, Emily B. Fox, and Erik B. Sudderth. Effective split-merge monte carlo methods for nonparametric models of sequential data. In *NeurIPS*, 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/hash/c5ab0bc60ac7929182aadd08703f1ec6-Abstract.html.

Sonia Jain and Radford M. Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182, 2004. doi: 10.1198/1061860043001.

J. F. C. Kingman. The representation of partition structures. *Journal of the London Mathematical Society*, 18(2):374–380, 1978. doi: 10.1112/jlms/s2-18.2.374.

Wouter Kool, Herke van Hoof, and Max Welling. Stochastic beams and where to find them: the gumbel-top-k trick for sampling sequences without replacement. In *ICML*, volume 97 of *PMLR*, pp. 3499–3508, 2019. URL https://proceedings.mlr.press/v97/kool19a.html.

Alex Kulesza and Ben Taskar. *Determinantal Point Processes for Machine Learning*. Now Publishers, 2012. doi: 10.1561/2200000044.

Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential monte carlo. In *ICLR*, 2018. URL https://openreview.net/forum?id=Byxm3q4K-.

Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998. doi: 10.1080/01621459.1998.10473765.

Chris J. Maddison, Dieterich Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Whye Teh. Filtering variational objectives. In *NeurIPS*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/fa84632d742f2729dc32ce8cb5d49733-Abstract.html.

Hongzi Mao, Malte Schwarzkopf, Shaileshh Bojja Venkatakrishnan, Zili Meng, and Mohammad Alizadeh. Learning scheduling algorithms for data processing clusters. In *ACM SIGCOMM*, pp. 270–288, 2019. doi: 10.1145/3341302.3342080.

Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B*, 79(4):1119–1141, 2017. doi: 10.1111/rssb.12200.

Rui Min, Christelle Garnier, François Septier, and John Klein. State space partitioning based on constrained spectral clustering for block particle filtering. In *arXiv preprint arXiv:2203.03475*, 2022. URL https://arxiv.org/abs/2203.03475.

Antonio Khalil Moretti, Liyi Zhang, Christian A. Naesseth, Hadiah Venner, David Blei, and Itsik Pe'er. Variational combinatorial sequential monte carlo methods for bayesian phylogenetic inference. In *UAI*, volume 161 of *PMLR*, pp. 971–981, 2021. URL https://proceedings.mlr.press/v161/moretti21a.html.

Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In *ICML*, volume 97 of *PMLR*, pp. 4663–4673, 2019. URL https://proceedings.mlr.press/v97/murphy19a.html.

Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *AISTATS*, volume 84 of *PMLR*, pp. 968–977, 2018. URL https://proceedings.mlr.press/v84/naesseth18a.html.

Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön. Elements of sequential monte carlo. *Foundations and Trends in Machine Learning*, 12(3):307–392, 2019. doi: 10.1561/2200000074.

Peter Orbanz and Daniel M. Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):437–461, 2015. doi: 10.1109/TPAMI.2014.2334607.

Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000. doi: 10.1080/01621459.2000.10473909.

Sally Paganin, Amy H. Herring, Andrew F. Olshan, and David B. Dunson. Centered partition processes: informative priors for clustering. *Bayesian Analysis*, 16(1):301–370, 2021. doi: 10.1214/20-BA1197.

Tiago P. Peixoto. Merge-split markov chain monte carlo for community detection. *Physical Review E*, 102:012305, 2020. doi: 10.1103/PhysRevE.102.012305.

Jim Pitman. *Combinatorial Stochastic Processes*. Springer, 2006. doi: 10.1007/b11601500.

Michael K. Pitt and Neil Shephard. Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999. doi: 10.1080/01621459.1999.10474153.

Gregory J. Puleo and Olgica Milenkovic. Correlation clustering with constrained cluster sizes and extended weights bounds. *SIAM Journal on Optimization*, 26(3):1857–1872, 2016. doi: 10.1137/140994198.

Aurick Qiao, Sang Keun Choe, Suhas Jayaram Subramanya, Willie Neiswanger, Qirong Ho, Hao Zhang, Gregory R. Ganger, and Eric P. Xing. Pollux: co-adaptive cluster scheduling for goodput-optimized deep learning. In *OSDI*, pp. 1–18, 2021. URL https://www.usenix.org/conference/osdi21/presentation/qiao.

Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Randy H. Katz, and Michael A. Kozuch. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In *SoCC*, 2012. doi: 10.1145/2391229.2391236.

Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008. doi: 10.1175/2008MWR2529.1.

Connie Trojan, Pavel Myshkov, Paul Fearnhead, James Hensman, Tom Minka, and Christopher Nemeth. Scalable model-based clustering with sequential monte carlo. In *AISTATS*, volume 300 of *PMLR*, pp. 3529–3537, 2026. URL https://proceedings.mlr.press/v300/trojan26a.html.

Yener Ulker, Bilge Günsel, and Taylan Cemgil. Sequential monte carlo samplers for dirichlet process mixtures. In *AISTATS*, volume 9 of *PMLR*, pp. 876–883, 2010. URL https://proceedings.mlr.press/v9/ulker10a.html.

Abhishek Verma, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. Large-scale cluster management at google with borg. In *EuroSys*, pp. 1–17, 2015. doi: 10.1145/2741948.2741964.

Sara Wade and Zoubin Ghahramani. Bayesian cluster analysis: point estimation and credible balls. *Bayesian Analysis*, 13(2):559–626, 2018. doi: 10.1214/17-BA1073.

Wencong Xiao, Romil Bhardwaj, Ramachandran Ramjee, Muthian Sivathanu, Nipun Kwatra, Zhenhua Han, Pratyush Patel, Xuan Peng, Hanyu Zhao, Quan Zhang, et al. Gandiva: introspective cluster scheduling for deep learning. In *OSDI*, pp. 595–610, 2018. URL https://www.usenix.org/conference/osdi18/presentation/xiao.

Kevin S. Xu and Alfred O. Hero. Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):552–562, 2014. doi: 10.1109/JSTSP.2014.2310294.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. In *NeurIPS*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

A Proof details

Support and importance correction. For every prior-supported neighbor Π', Equation (5) gives $q_\lambda(\Pi' \mid \Pi, y) \geq \lambda p(\Pi' \mid \Pi) > 0$. For any bounded f, conditioning on an ancestor particle yields

$$\mathbb{E}_{q_\lambda}\left[f(\Pi')\frac{p(\Pi' \mid \Pi)p(y \mid \Pi')}{q_\lambda(\Pi' \mid \Pi, y)}\right] = \sum_{\Pi'} f(\Pi')p(\Pi' \mid \Pi)p(y \mid \Pi'),$$

which is the unnormalized filtering update. Normalization and standard resampling therefore target the intended posterior. Tests enumerate all neighbors for small partitions, confirm that p and q each sum to one, and compare returned probabilities with an independent evaluator.

Equivariance. A relabeling maps blocks bijectively while preserving their sizes. Residual sums and merge pairs therefore map bijectively with unchanged logits. For a split, the key-defined anchor maps to the anchor of the relabeled block. The multiset of subset weights and every ESP coefficient are unchanged, so the probability of A equals that of σA. The prior mixture has the same property. Composition with resampling consequently gives an equivariant particle distribution.

Complexity. Node residuals and candidate target sizes require $O(k|V|)$ work in the direct implementation; pairwise merge scores require $O(k^2)$. For each candidate split block, the ESP table is quadratic in block size, summing to $O(\sum_C |C|^2)$. No step constructs the $2^{|C|-1} - 1$ splits of C.

B Protocol and additional results

Table 6: Multi-step results. Each row contains 100 paired episodes.

delay/loss/noise	PRPT obj.	bootstrap	robust	stale	oracle
1/0.0/0.0	116.25	112.49	106.81	83.70	127.50
2/0.1/0.25	106.29	99.28	84.80	40.15	127.50
2/0.1/0.5	105.86	100.20	82.18	40.08	127.50
3/0.2/0.5	94.86	86.86	67.08	-2.24	127.50
4/0.3/0.75	85.99	76.84	53.85	-27.20	127.50
2/0.1/1.0	100.25	95.66	78.10	40.18	127.50

The Alibaba construction masks current server identity before inference. Each episode uses a lagged membership partition, a request-derived pressure vector available at decision time, the observed next partition as label, and an explicitly enumerated one-step posterior for evaluation. Calibration uses days 30/60 only; the configuration is frozen before days 170/180 and again before the predeclared non-exact days 160/175/184 are loaded. Five-day pooling is reported only as descriptive robustness because strata follow different inclusion rules.

C ARTIFACT MAP

The supplementary repository separates immutable raw results from rendering scripts. The exact, factorized, multi-step, random-topology, independent-domain, Alibaba, and timing JSON files include seeds and configurations. The final evidence manifest records SHA-256 hashes for the two real-data result sets, runner and core sources, input Parquet shards, and timing reruns. Paper tables are rebuilt from compact exported JSON/CSV rather than manual transcription.