FORECAST QUALITY AND RECOURSE IN SEMANTIC FACILITY PLACEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Forecasts guide the residency of service capabilities before demand is known, while compatible capacity may also be activated afterward at a recourse cost. We show that this distinction changes both the value of anticipation and the appropriate learning target. In the stated two-stage model, newly planned activations can be deferred without changing assignments, physical states, or total cost when emergency activation has no extra fee or delay. With paid recourse, a separable capability admits a Bayes one-step rule based on expected service savings clipped by capacity and activation cost. The conditional mean of demand generally does not determine that rule. A squared-loss calibration bound motivates learning the clipped value directly. On 12 matched generated conditions, selecting forecasts by local-search rollout cost reduces cost by 2.05% relative to EMA, but the improvement does not transfer to the same forecasts under MPC. A separate replay uses 104 structured API calls from 40 public annotated dialogues under 54 explicit cost mappings. The clipped target changes cost in only six mappings.

1 INTRODUCTION

Demand forecasts inform which service capabilities remain resident between requests. A tool service, model endpoint, or data-access bundle incurs a recurring cost while active and a transition cost when its state changes. After demand arrives, the system may activate additional compatible capacity, route work remotely, or reject it. The economic value of forecasting therefore depends on what these later decisions can recover and what they cost.

This dependence is easy to overlook in a placement simulator. If an emergency activation has the same cost and availability as a planned activation, and neither incurs a delay, why activate a previously closed facility before observing demand? We show that it is unnecessary in the stated model. Every feasible early-activation execution has a deferred counterpart with the same assignments, final state, and total cost. Forecasts can still inform whether an existing bundle should remain active. However, an observed advantage of an anticipatory heuristic cannot, by itself, establish that advance activation has intrinsic value.

Once emergency activation carries an additional charge, the prediction target changes. For a single optional capability, the relevant random quantity is the local-service saving clipped by both finite capacity and the cost of emergency activation. Its conditional mean, rather than the conditional mean of demand, determines the optimal one-step decision. The previous physical state sets a different threshold for opening and retaining the capability. A mean-count predictor can systematically overestimate this clipped value, even when it minimizes count mean-squared error. This identifies a specific recourse mechanism behind the familiar distinction between prediction accuracy and decision quality.

We study the mechanism in semantic facility placement. Here *semantic* means a given compatibility relation between task types and service bundles. It does not mean learned language similarity or inferred permissions. At one epoch, a fixed mask yields a capacitated facility-location problem on a restricted assignment graph. Our contribution is not a new complexity classification for that model. It is an analysis of the decision target induced by its two-stage execution costs, together with matched experiments that separate forecast training, placement search, and recourse.

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.

The paper has three main results. First, a pathwise deferral identity identifies the assumptions under which early activation has no additional value. Second, a recourse-clipped conditional target gives a Bayes one-step activation rule and a squared-loss calibration bound for separable capabilities. Third, a crossed experimental design compares count-based and cost-based forecast selection through the same local and rolling-horizon optimizers. The evaluation distinguishes achieved optimization quality from the requested solver budget and includes a small replay of public annotated API calls under multiple explicit cost mappings. The replay is an external input to a counterfactual placement model, not a production-serving experiment.

Relation to prior work. Prediction error need not align with decision loss, as established by Smart Predict-then-Optimize and decision-focused learning (Elmachtoub and Grigas, 2022; Wilder et al., 2019). We build on that distinction rather than claim it as new. Our analytical result identifies the clipped target created by emergency activation and persistent state. Our empirical question is whether choosing a forecast by one controller's cost transfers to another controller. It need not, even when their execution accounting agrees.

Classical facility location and its online variants study opening and assignment costs (Shmoys et al., 1997; Jain and Vazirani, 2001; Meyerson, 2001). Prediction-augmented facility location and multiple advice sources are also established (Almanza et al., 2021; Fotakis et al., 2025). The prediction target, recurring residency cost, and two-stage recourse here differ from those settings. No competitive ratio from those works is transferred to our heuristic. Rolling-horizon mixed-integer control is a substantive comparator, following the model-predictive-control principle of optimizing a forecast trajectory and executing its first action (Rawlings et al., 2017).

Network embedding includes resource and service dependencies beyond our model (Chowdhury and Boutaba, 2010; Fischer et al., 2013). Tool-use benchmarks supply concrete capability identities (Li et al., 2023; Qin et al., 2024; Trivedi et al., 2024; Fan et al., 2025), while Orca, vLLM, SGLang, and RouteLLM address serving or model selection (Yu et al., 2022; Kwon et al., 2023; Zheng et al., 2024; Ong et al., 2024). We neither implement those systems nor infer their performance from a placement-cost table. The present study concerns a slower residency decision under a specified execution model.

2 Placement, Assignment, and Paid Repair

Let $\mathcal{A}$ be a finite set of task types and $\mathcal{F}$ a finite set of candidate bundles. Type a has nonnegative resource vector $d_a \in \mathbb{R}^D_+$, and bundle i has capacity $U_i \in \mathbb{R}^D_+$. The given mask $K_{ai} \in \{0,1\}$ states whether i may serve a. In the experiments it combines family membership, permission level, and a region predicate. It is fixed within each run. The model does not infer permissions from data.

Let $f_i \geq 0$ be the per-epoch residency cost, $m_i \geq 0$ the cost of toggling bundle i, $c_{ai} \geq 0$ its connection cost, and $R_a > 0$ the penalty for rejected mass. An emergency activation may also incur surcharge $\gamma \geq 0$, which is zero outside the surcharge sweep. Define

$$M(x, s) = \sum_i m_i |x_i - s_i|. \tag{1}$$

This is a weighted Hamming pseudometric on binary placements. A toggle is not a source-to-destination transfer of an owned memory shard. Such a transfer would require additional state-conservation variables that are absent from this study.

At epoch t, the controller observes the previous physical placement q_{t-1} and past demands, but not n_t. It selects a planned placement p_t. After n_t is observed, assignment may activate additional bundles, producing $q_t \geq p_t$. Accepted and rejected masses are $y_{ai,t} \geq 0$ and $z_{a,t} \geq 0$. The realized

one-epoch cost and feasible region are

$$
\begin{aligned}
C_t &= f^\top q_t + M(p_t, q_{t-1}) + M(q_t, p_t) \\
&\quad + \sum_{a,i} c_{ai} y_{ai,t} + \sum_a R_a z_{a,t} + \gamma \mathbf{1}^\top (q_t - p_t), \\
\sum_i y_{ai,t} + z_{a,t} &= n_{a,t} && \forall a, \\
\sum_a d_a^k y_{ai,t} &\le U_i^k q_{i,t} && \forall i, k, \\
y_{ai,t} &\le n_{a,t} q_{i,t} && \forall a, i, \\
y_{ai,t} &= 0 && \text{when } K_{ai} = 0, \\
p_t, q_t &\in \{0,1\}^{|\mathcal{F}|}, \quad q_t \ge p_t, \quad y_t, z_t \ge 0.
\end{aligned}
\tag{2}
$$

The controller pays both transitions. Closing and reopening the same bundle within an epoch does not cancel the charges. Every activated bundle persists as part of q_t until a later planned decision closes it. No policy receives a special reset of its fallback facilities.

Rejection is an explicit outcome, not a successful assignment. It ensures a feasible decision exists even if no compatible capacity remains. Repair means activating a preapproved candidate bundle. It does not mean creating a new permission domain or unlimited capacity. The simulator assumes that activation completes within the epoch and charges its cost but not a startup delay. This assumption limits any interpretation of modeled connection costs as service times.

Aggregate masses may be fractional. The resource inequalities describe per-epoch work and occupancy units rather than an event-level queue. In particular, they do not establish simultaneous memory feasibility for a running autoregressive model. We report rejection, emergency service, and connection cost separately from the objective.

3 Forecast selection and matched controllers

For forecast $\hat{n}_t$, uniform reserve $\epsilon \in [0, 1)$, and movement weight $\lambda \ge 0$, the planning problem is

$$
\begin{aligned}
\min_{x,y,z} \quad & f^\top x + \lambda M(x, q_{t-1}) + \sum_{a,i} c_{ai} y_{ai} + \sum_a R_a z_a \\
\text{subject to} \quad & \sum_i y_{ai} + z_a = \hat{n}_{a,t} && \forall a, \\
& \sum_a d_a^k y_{ai} \le (1-\epsilon) U_i^k x_i && \forall i, k, \\
& y_{ai} \le \hat{n}_{a,t} x_i && \forall a, i, \\
& y_{ai} = 0 && \text{when } K_{ai} = 0, \\
& x \in \{0,1\}^{|\mathcal{F}|}, \quad y, z \ge 0.
\end{aligned}
\tag{3}
$$

The reserve discounts every facility-resource pair uniformly. It does not provide type-specific coverage when a compatible facility was never opened. Rejection is part of the objective and demand conservation, so the all-zero decision cannot serve positive demand at zero cost.

Local search and MPC. The local planner compares the previous physical placement with capacity-cover constructions initialized from the previous and empty placements. It then tests deletion of four low-utilization bundles and accepts only decreases in its deterministic assignment score. This is a bounded heuristic without a proved approximation factor. MPC repeats Eq. (3) over three recursively forecast epochs, couples consecutive placements by toggle costs, and executes only the first planned placement. Both use $\epsilon = 0.07$, $\lambda = 0.95$, identical candidate facilities, and the same realized assignment and repair rule.

The primary MPC comparison uses a 64-node budget without a wall-clock cap, no warm start, and relative gap target 10^{-7}. We compare achieved optimality gaps and termination status under

Algorithm 1 Solver-matched forecast calibration and evaluation

Fit ridge coefficients on $\mathcal{D}_{\text{fit}}$ and form the fixed grid Γ
for $g \in \Gamma$ **do**
  $L_N(g) \leftarrow \sum_{t \in \mathcal{D}_{\text{cal}}} \|n_t - \hat{n}_t^g\|_2^2$
  for $S \in \{\text{Local}, \text{MPC}\}$ **do**
    Reset $q_0 \leftarrow 0$ and roll out S on $\mathcal{D}_{\text{cal}}$ with forecast g
    $L_S(g) \leftarrow \sum_{t \in \mathcal{D}_{\text{cal}}} C_t$ under Eq. (2)
  end for
end for
Freeze $g_N \leftarrow \arg\min_g L_N(g)$ and $g_S \leftarrow \arg\min_g L_S(g)$
Evaluate EMA, ridge, g_N, g_{Local}, and g_{MPC} through both solvers
Use disjoint test demand, a common all-closed initial state, and common repair rules

fixed-node and clock-limited budgets. A solve without a feasible incumbent falls back to the same local planner.

Forecast fitting and calibration. A pooled ridge autoregression predicts each type from six lagged counts, its training mean, and a lag difference. A six-lag EMA is the nonlearned reference. The ridge coefficients are fitted on the first 96 training epochs. The following 24 epochs are reserved for calibration. Neither model fitting nor configuration selection uses the test stream. The nine candidate forecasts are

$$\hat{n}^{\alpha,v} = v\big[(1-\alpha)\hat{n}^{\text{EMA}} + \alpha\hat{n}^{\text{ridge}}\big], \qquad \alpha \in \{0, \tfrac{1}{2}, 1\}, \quad v \in \{0.8, 1, 1.2\}. \tag{4}$$

Count calibration minimizes calibration mean-squared error. Cost calibration minimizes the full calibration rollout cost using a specified downstream solver. The selected configuration is frozen on test. We evaluate both cost-calibrated choices through both solvers, so forecast selection and optimization can be distinguished.

For comparison with the broader generated suites, we retain SAGE-R, the 24-slot historical portfolio over four forecasts, three reserves, and two movement weights. Its ablations retain the same scoring budget, including duplicate configurations after component removal. Historical scores divide by $\max\{1, \mathbf{1}^\top n_t\}$, which defines zero-demand epochs. This portfolio is not an independent-expert Hedge process. Its full specification is in Appendix E.

4 How paid recourse changes the prediction target

The execution model permits activation after demand is observed. Its cost must therefore be compared with early activation before interpreting an anticipatory controller's performance. The following result applies to coupled capacities and arbitrary compatibility masks. Proofs are in Appendix A. We write $C(p, q, y, z \mid s)$ for the cost in Eq. (2), and $|p \setminus s| = \sum_i p_i(1 - s_i)$ for the number of newly planned activations.

Theorem 1 (Deferral identity)**.** *Fix a previous binary state s and a feasible execution (p, q, y, z) of Eq. (2). Let $\tilde{p} = p \wedge s$ be the coordinatewise minimum. The execution $(\tilde{p}, q, y, z)$ is feasible and*

$$C(\tilde{p}, q, y, z \mid s) - C(p, q, y, z \mid s) = \gamma\,|p \setminus s|. \tag{5}$$

In particular, when $\gamma = 0$, newly planned activations can be deferred without changing assignments, final physical state, or cost. Applying this construction causally at every epoch preserves the full state trajectory and its total cost.

The theorem assumes that delayed activation remains available in the same epoch and incurs no startup delay. It preserves planned retirements and changes only newly activated bundles. Thus it does not remove the forecasting problem for retention. Nor does it imply that a different greedy repair routine will discover the deferred execution. It identifies a property of the modeled feasible decisions. With $\gamma > 0$, deferral can incur an additional fee, while avoiding an unnecessary activation can still be beneficial.

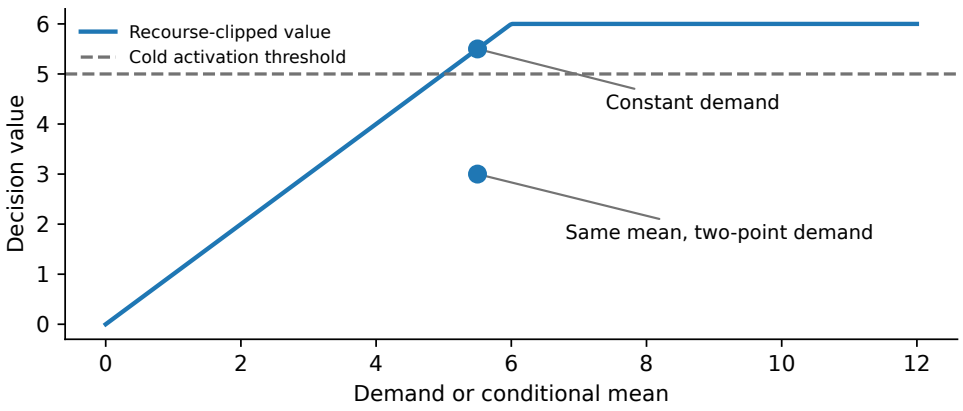

Figure 1. The same conditional mean can imply different activation decisions. Constant demand and a two-point distribution both have mean 5.5, but their expected clipped values lie on different sides of the threshold. The curve is the deterministic transform, not a fitted experimental response.

A separable capability model. Consider one optional local capability with capacity U, recurring cost f, toggle cost m, and emergency surcharge γ. A compatible remote service has unlimited capacity and unit cost b. Local unit cost is $c < b$. Demand $N \geq 0$ is integrable. Let $q \in \{0, 1\}$ be the previous state and define

$$s(N) = (b-c)\min\{N, U\}, \quad h = f + m + \gamma, \quad T(N) = \min\{s(N), h\}, \quad k_q = f + m(1-2q). \tag{6}$$

The local and remote services are compatible with the same task. Remote service is not rejection or a policy exception. This restricted model isolates the retention and repair thresholds without coupling different types through shared capacities.

Theorem 2 (Recourse value and retention thresholds)**.** *Under the separable model, the minimum realized cost conditional on choosing early activation differs from the minimum cost conditional on waiting by $k_q - T(N)$. Given pre-decision information $\mathcal{H}$, an optimal one-step decision activates or retains the capability precisely when*

$$\mu(\mathcal{H}) = \mathbb{E}[T(N) \mid \mathcal{H}] > k_q, \tag{7}$$

with ties resolved by waiting. The cold activation threshold is $f + m$, whereas the retention threshold is $f - m$.

The economic target is clipped twice. Capacity bounds how much local service can save, and affordable emergency activation bounds the cost of waiting. Since T is concave, $T(\mathbb{E}[N \mid \mathcal{H}]) \geq \mathbb{E}[T(N) \mid \mathcal{H}]$. A conditional mean-count forecast therefore overestimates the relevant value when passed through the deterministic planner. The two quantities coincide on an affine segment of T, but may differ when demand straddles a capacity or repair threshold.

For example, take $b - c = 1$, $U = 12$, $f = 4$, $m = 1$, $\gamma = 1$, and $q = 0$. Constant demand $N = 5.5$ and a distribution placing equal mass on 0 and 11 have the same mean. Their expected clipped values are 5.5 and 3, respectively, on opposite sides of the activation threshold 5. A mean-count predictor cannot distinguish these decisions. With no emergency surcharge and a cold initial state, $T(N) \leq h = k_0$, so waiting is always a one-step minimizer, consistent with Theorem 1.

Proposition 3 (Decision-aligned regression)**.** *Let $\hat{\mu}(\mathcal{H})$ estimate $\mathbb{E}[T(N) \mid \mathcal{H}]$ and use the threshold rule in Eq. (7). Its conditional excess one-step cost is*

$$|\mu - k_q| \, \mathbf{1}\{(\hat{\mu} > k_q) \neq (\mu > k_q)\} \leq |\hat{\mu} - \mu|. \tag{8}$$

Consequently, its expected excess cost is at most the square root of its excess squared prediction risk for target $T(N)$.

Continuation value. Persistent state also affects future epochs. For an exogenous demand process, let $V_{t+1}(r, \mathcal{H}_{t+1})$ be the optimal remaining cost from physical state r and let $D_t = V_{t+1}(1, \mathcal{H}_{t+1}) -$

Table 1. Forecasts crossed with downstream solvers on the 12 held-out medium conditions. Positive percentages favor the displayed forecast over EMA using the same solver. Brackets give within-family bootstrap 95% intervals. Calibration minimizes either count MSE or the named solver's rollout cost. Bold marks the highest displayed mean reduction within a solver.

Forecast	Local search	MPC
Count regression	-1.66 [-3.07, -0.43]	**+1.14** [+0.45, +1.91]
Count-selected grid	-0.90 [-1.50, -0.27]	+0.87 [+0.22, +1.61]
Cost-selected local	**+2.05** [+0.75, +3.54]	+0.05 [-0.49, +0.55]
Cost-selected MPC	-2.04 [-3.32, -0.57]	+0.71 [+0.10, +1.36]

$V_{t+1}(0, \mathcal{H}_{t+1})$. The finite-horizon single-capability rule replaces μ by

$$\mathbb{E}[\min\{s(N_t) - D_t, h\} \mid \mathcal{H}_t]. \tag{9}$$

Appendix A.5 derives this identity by comparing the two Bellman branches. The one-step clipped target is recovered at the terminal epoch, when $D_t = 0$. At zero surcharge the cold-state target still cannot exceed $h = k_0$, but retention can depend on future use. Thus learning the correct one-step target and valuing a persistent state over a horizon are different problems. The replay reports a full-horizon reference to keep that distinction visible.

This gives a training target for the separable decision, not a global regret guarantee or an approximation theorem for local search. Shared capacities generally destroy separability. We therefore test direct value regression only in the separable replay, and use solver-specific cost calibration in the coupled study. The posterior LP bounds for the coupled model remain useful diagnostics, but their weak-duality argument is not the main contribution. Their assumptions, numerical qualifications, and distinction from trajectory regret appear in Appendix B.

5 Experiments

The experiments separate three questions. Does the recourse model itself reward advance activation? Does forecast selection transfer between optimizers? Does the clipped prediction target change decisions on independently supplied tool-call sequences? Generated scenarios, annotated-call replay, and hardware-calibrated coefficients are reported separately.

Coupled instances and held-out selection. Twelve medium conditions combine four generated scenario families with three seeds each. Each condition has 12 task types, 24 ordinary bundles, three finite fallback bundles, and 24 test epochs of 500 offered units. Counts, permissions, regions, costs, and capacities are generated. The families use labels inspired by tool-use benchmarks but are not executions of those benchmarks. The training stream has 96 fitting epochs followed by 24 calibration epochs, and the test stream uses a disjoint seed. All ten test arms share the execution rule, initial state, and offered demand. The four calibrated arms use the same nine candidate forecasts. These choices are fixed before the test outcomes are computed.

For a policy pair, improvement is $100(1 - C_A/C_B)$ within each family-seed condition. Means weight the four families equally. Intervals use 20,000 seed resamples within each family while holding the family set fixed. They do not quantify uncertainty over a population of deployed systems. All negative per-condition results are retained.

Solver-specific calibration. Ridge reduces one-step count L_1 error relative to EMA by 15.06%, with interval $[13.81, 16.36]$. Its local-search cost nevertheless increases by 1.66%, while its MPC cost decreases by 1.14% (Table 1). Selecting a forecast by local-search calibration cost reverses the local result, yielding a 2.05% mean reduction relative to EMA. That selected forecast gives only a 0.05% reduction under MPC, with an interval crossing zero. Conversely, MPC-selected calibration worsens local-search cost by 2.04% and improves MPC cost by 0.71%. The comparison therefore supports solver-specific selection, not a universally superior forecaster. It also does not establish that cost calibration dominates ridge under MPC.

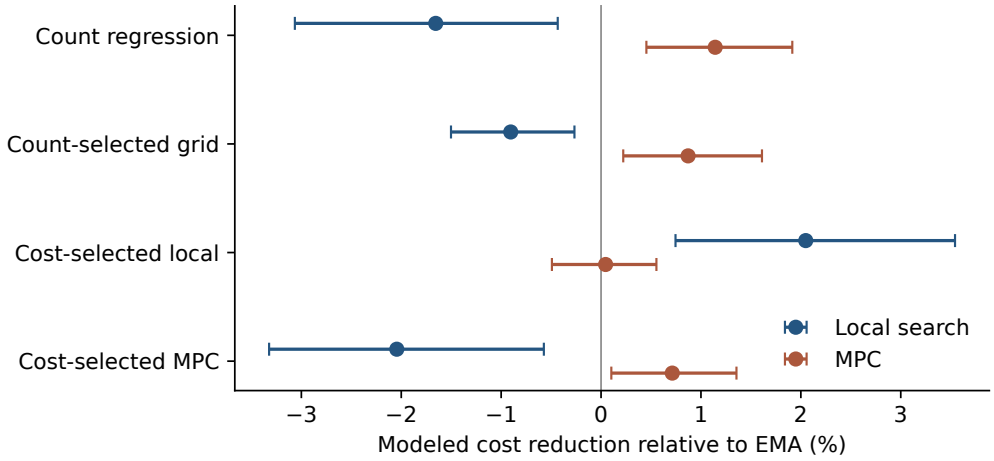

Figure 2. Calibration does not transfer uniformly between placement solvers. Points and intervals use the same paired conditions as Table 1. Choosing forecasts by one solver's cost can worsen another solver's decisions.

Achieved optimization quality. All 1,440 primary test MPC calls returned solver status optimal, with maximum reported relative gap 2.3×10^{-14} and no local fallback. These are numerical solver outcomes, not real-arithmetic certificates. We then replayed 96 common states with three frozen forecasts under a 64-node budget alone, the same budget plus 0.5 seconds, and the same budget plus 0.05 seconds. Each clock-capped input was run twice. All 288 node-only and 576 half-second calls terminated optimally. At 0.05 seconds, time limits occurred in 4 of 192 EMA calls, 16 of 192 ridge calls, and 4 of 192 MPC-calibrated calls. Maximum gaps were 1.29%, 80.88%, and 5.25%, respectively. Thus an equal clock cap need not give equal achieved optimization quality. Appendix D.2 gives the full status, incumbent, bound, timing, and fallback definitions.

Deferral counterfactual. On those same 96 states, we compare a local EMA plan with its deferred version under optimal one-epoch recourse at surcharges 0, 20, and 100. Forecasts and the previous physical state are held fixed. The deferred decision never costs more at zero surcharge, as Theorem 1 requires. Its mean modeled cost is 3.30 units lower because it can avoid unnecessary activations. At surcharges 20 and 100, the mean difference changes to 2.62 and 22.77 units higher. Every paired difference lies below $\gamma|p \setminus s|$. All 576 recourse solves have reported gaps below 8.0×10^{-15}. These are controlled one-step comparisons, not complete alternative policy rollouts.

Public annotated-call replay. We use a fixed subset of 40 public API-Bank dialogues (Li et al., 2023). The subset contains 104 annotated API calls, 26 API identities, and 38 distinct call sequences. Only explicit API events contribute demand. Actions mentioned only in task labels or dialogue text do not. One dialogue with no annotated calls is retained as a zero-demand case. Consecutive blocks of at most two calls form epochs within each dialogue, preserving their order. The annotations provide no measured inter-arrival times, service durations, or production placement states.

The separable replay assigns each API a compatible local capability and a compatible remote service. Fifty-four fixed mappings cross two local capacities, three residency costs, three toggle costs, and three emergency surcharges. These are counterfactual resource prices rather than measured API costs. Count and clipped-value regressors have the same last-API context, smoothing strength, and five-fold split. Identical projected sequences are kept in the same fold. Each held-out dialogue is evaluated once per mapping, with no test-dependent selection of a price or model.

The clipped target improves cost in six mappings and leaves it unchanged in 48, for a mean reduction of 0.25% and range $[0, 3.56]\%$ (Table 2). There are no observed losses in this small subset, but that is not a dominance guarantee. The limited effect is informative. Clipping changes the learned decision only in part of the tested threshold range. The other mappings provide no evidence of a practical advantage. A common-state diagnostic finds 86 action disagreements across all mapping replays.

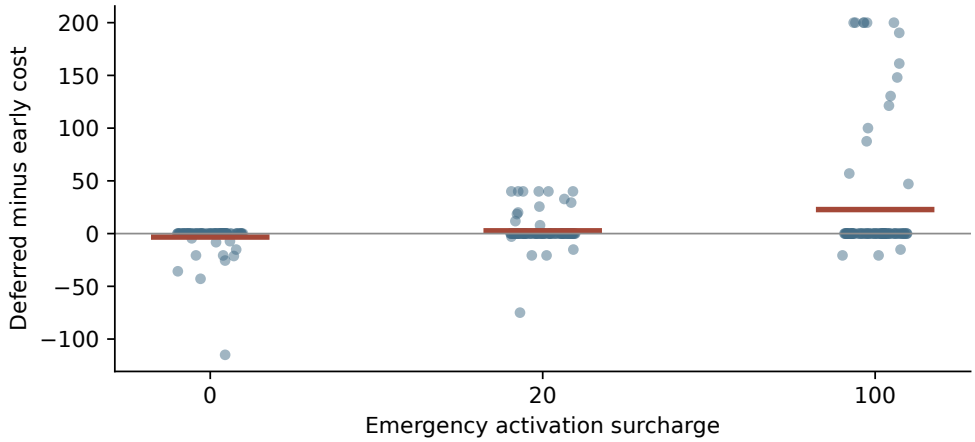

Figure 3. Fixed-state deferral differences. Small points are the 96 states at each fee, and horizontal marks are means. At zero surcharge, delayed optimal recourse never costs more. Higher fees change the tradeoff without guaranteeing that every early activation is useful.

Table 2. Clipped-value versus count regression on the annotated-call replay. Gain is the mean of per-mapping percentage reductions in total held-out modeled cost. The range retains ties. Mappings reuse the same dialogues and are not independent samples.

Surcharge	Mappings	Mean gain	Range	Wins / ties
0	18	0.36%	[0.00, 2.72]	4 / 14
0.5	18	0.39%	[0.00, 3.56]	2 / 16
2	18	0.00%	[0.00, 0.00]	0 / 18

Twenty occur at zero surcharge and concern retention of an already active capability, never advance activation. The other 66 occur from a cold state with positive surcharge. Of these 86 one-step changes, 82 reduce realized cost and four increase it. This diagnostic holds the count policy's previous state fixed and is not a comparison of complete state trajectories. The replay uses externally authored call identities and order, while capacities and prices remain modeled inputs. It does not validate a deployed agent scheduler.

Broader scenarios and accelerator calibration. The larger generated suites remain in Appendix F, including all 824 controller runs, eight compact full-horizon solves, and seven independent sensitivity axes. Their matched historical-portfolio gain over fixed EMA is 0.21%, with interval $[-0.49, 0.88]$, so they do not establish a reliable portfolio advantage. The A100-calibrated simulation has a 1.55% modeled-cost reduction over EMA, with interval $[0.64, 2.69]$, but higher emergency-served mass and similar rejection. Its kernel rates are inputs to a hand-chosen coefficient mapping. No live model-serving, startup-delay, or end-to-end GPU experiment is claimed. These results complement the mechanism study without turning simulated demand units into independent evidence.

6 Discussion and Limitations

The central result concerns the value of recourse, not the general usefulness of forecasting. When a simulator permits immediate activation after observing demand at no additional fee, advance activation can be deferred at equal cost. A predictive controller may still improve retention, candidate search, or the use of shared resources. Those effects must not be attributed automatically to early provisioning. Introducing a surcharge changes the decision through a clipped service-value target and separate opening and retention thresholds.

The one-capability analysis is intentionally separable. Shared capacities, workflow precedence, startup delay, and conserved memory objects can change both the feasible recourse and the optimal

prediction target. Appendix A.5 states the continuation-value correction for a finite-horizon single-capability problem. Neither that identity nor the posterior LP diagnostics supplies a global guarantee for the implemented coupled controller. Developing an efficient coupled approximation to the continuation value is a more direct next step than attaching a generic competitive bound to local search.

The external replay is small and counterfactual. Its call identities and within-dialogue order come from public annotations, but it does not provide production arrival rates, resource measurements, or observed permission changes. The fixed 54 mappings expose this uncertainty rather than fit it away. Their small gains should not be extrapolated to deployed systems. A stronger systems study would measure the actual cost and delay of activating a capability, then replay observed demand through competing policies with the same state persistence and admission rule.

7 Conclusion

Paid recourse determines which forecast errors matter for capability placement. The deferral identity exposes a zero-surcharge boundary at which advance activation has no intrinsic cost advantage. In a separable model, the appropriate one-step target is expected recourse-clipped service value, with a threshold that depends on the previous physical state. Matched experiments show that cost-calibrated forecasts need not transfer between placement solvers, while a small annotated-call replay locates the limited range in which the clipped target changes decisions. The resulting claims are about a specified optimization model and its tested decision mechanisms, not a general serving-speed or deployment guarantee.

AI USE STATEMENT

Generative AI assisted in literature search, grammatical and structure revision, and figure construction. The authors assume responsibility for the content.

ETHICS STATEMENT

No new human-subject data or private service logs are collected. The experiments use generated workloads, a supplied accelerator calibration, and a projection of public annotated API calls containing only tool identities and their order. No user utterances, credentials, or argument values are redistributed in the projection. Capability and permission masks are abstract inputs, not a certification of compliance with a legal or organizational policy. A deployment would require authoritative permission checks, accounting for startup delays, and explicit treatment of rejected requests.

REPRODUCIBILITY STATEMENT

Proofs, experimental settings, and additional results appear in the appendices. The code and data are available in the supplement.

REFERENCES

Matteo Almanza, Flavio Chierichetti, Silvio Lattanzi, Alessandro Panconesi, and Giuseppe Re. Online Facility Location with Multiple Advice. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/250473494b245120a7eaf8b2e6b1f17c-Abstract.html.

N. M. Mosharaf Kabir Chowdhury and Raouf Boutaba. A Survey of Network Virtualization. *Computer Networks*, 54(5):862–876, 2010.

Adam N. Elmachtoub and Paul Grigas. Smart "Predict, then Optimize". *Management Science*, 68(1): 9–26, 2022. doi: 10.1287/mnsc.2020.3922. URL https://arxiv.org/abs/1710.08005.

Shiqing Fan, Xichen Ding, Liang Zhang, and Linjian Mo. MCPToolBench++: A Large Scale AI Agent Model Context Protocol MCP Tool Use Benchmark. *arXiv preprint arXiv:2508.07575*, 2025.

Andreas Fischer, Juan Felipe Botero, Michael Till Beck, Hermann de Meer, and Xavier Hesselbach. Virtual Network Embedding: A Survey. *IEEE Communications Surveys & Tutorials*, 15(4): 1888–1906, 2013.

Dimitris Fotakis, Evangelia Gergatsouli, Themistoklis Gouleakis, Nikolas Patris, and Thanos Tolias. Improved Bounds for Online Facility Location with Predictions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25):26973–26981, 2025. doi: 10.1609/aaai.v39i25.34903. URL https://ojs.aaai.org/index.php/AAAI/article/view/34903.

Kamal Jain and Vijay V. Vazirani. Approximation Algorithms for Metric Facility Location and k-Median Problems Using the Primal-Dual Schema and Lagrangian Relaxation. *Journal of the ACM*, 48(2):274–296, 2001.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *ACM Symposium on Operating Systems Principles*, 2023.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3102–3116, 2023. doi: 10.18653/v1/2023.emnlp-main.187. URL https://aclanthology.org/2023.emnlp-main.187/.

Adam Meyerson. Online Facility Location. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 426–431, 2001.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to Route LLMs with Preference Data, 2024. URL https://arxiv.org/abs/2406.18665.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-World APIs. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.16789.

James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2 edition, 2017. URL https://sites.engineering.ucsb.edu/~jbraw/mpc/. First printing, October 2017.

David B. Shmoys, Éva Tardos, and Karen Aardal. Approximation Algorithms for Facility Location Problems. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 265–274, 1997.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the Data-Decisions Pipeline: Decision-Focused Learning for Combinatorial Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1658–1665, 2019. doi: 10.1609/aaai.v33i01.33011658. URL https://ojs.aaai.org/index.php/AAAI/article/view/3982.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. Orca: A Distributed Serving System for Transformer-Based Generative Models. In *USENIX Symposium on Operating Systems Design and Implementation*, 2022.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient Execution of Structured Language Model Programs, 2024. URL https://arxiv.org/abs/2312.07104. Revised arXiv version, June 2024.

A RECOURSE MECHANISM AND PREDICTION TARGET

This appendix proves the main results under the stated two-stage model. The compatibility relation and capacities constrain the final assignments. An emergency opening can activate only an admissible candidate, and it completes within the current epoch. There is no separate startup delay, early-reservation constraint, or limit on the number of post-observation activations. These assumptions are essential to the deferral construction.

A.1 PROOF OF THEOREM 1

Proof. Write $s_i, p_i, q_i \in \{0,1\}$ with $q_i \geq p_i$ and set $\tilde{p}_i = s_i p_i$. Since $\tilde{p}_i \leq p_i \leq q_i$, the new planned placement and unchanged final placement satisfy the execution chronology. All assignment, compatibility, demand, and resource constraints depend on q, y, z, which are unchanged. Thus the deferred execution is feasible.

For a single coordinate, binary arithmetic gives

$$\begin{aligned} |p_i - s_i| + |q_i - p_i| &= p_i + s_i - 2p_i s_i + q_i - p_i \\ &= s_i + q_i - 2p_i s_i, \\ |\tilde{p}_i - s_i| + |q_i - \tilde{p}_i| &= s_i - s_i p_i + q_i - s_i p_i \\ &= s_i + q_i - 2p_i s_i. \end{aligned}$$

Multiplying by m_i and summing proves

$$M(\tilde{p}, s) + M(q, \tilde{p}) = M(p, s) + M(q, p). \tag{10}$$

In particular, a planned retirement followed by emergency reactivation remains charged twice. The construction does not cancel or remove that retirement. Residency, assignment, and rejection costs also remain unchanged. Only the surcharge changes, by

$$\gamma \sum_i \left[(q_i - \tilde{p}_i) - (q_i - p_i) \right] = \gamma \sum_i p_i (1 - s_i) = \gamma |p \setminus s|. \tag{11}$$

This proves Eq. (5).

For the pathwise statement, emulate any original policy causally. Before demand arrives, compute its virtual plan p_t using the observed history and state, but implement $p_t \wedge q_{t-1}$. After observing n_t, compute the original policy's final execution and use the same q_t, y_t, z_t. The preceding argument makes that execution feasible. The two policies have the same final state and observations at the end of the epoch, so induction preserves the emulation at the next epoch. When $\gamma = 0$, every epoch cost is equal. For positive surcharge, the emulation pays an additional $\gamma \sum_t |p_t \setminus q_{t-1}|$. An optimal deferred execution can be cheaper than this particular emulation because it is free to omit activations that the early plan made unnecessary. □

The result concerns the feasible decision model, not the exact output of a specific greedy repair subroutine. Changing a heuristic's planned set can change its subsequent search order. Consequently a difference between two such heuristic executions is compatible with the identity. The fixed-state counterfactual experiment instead optimizes both recourse problems to the reported solver gaps.

A.2 PROOF OF THEOREM 2

Proof. Let $d = b - c > 0$. The best assignment when the local capability is open serves $\min\{N, U\}$ locally and the rest remotely, saving $s(N) = d \min\{N, U\}$ relative to all-remote service. There is no reason to use less local capacity because all other costs have already been charged and $c < b$.

If the pre-demand action is on, the final state must be on. Its cost is

$$C_1(N, q) = bN - s(N) + f + m(1 - q). \tag{12}$$

If the pre-demand action is off, it pays mq to close a previously active capability. After demand arrives it either remains off at remote cost bN, or opens locally at additional fixed cost f, toggle cost m, and surcharge γ, while saving $s(N)$. Therefore

$$C_0(N, q) = bN + mq + \min\{0, f + m + \gamma - s(N)\}. \tag{13}$$

These two branches include all allowed final decisions. Define $B(N, q) = bN + mq$ and $k_q = f + m(1-2q)$. Then $C_1 = B + k_q - s(N)$ and $C_0 = B + \min\{0, h - s(N)\}$, where $h = f + m + \gamma$. Subtraction yields

$$C_1 - C_0 = k_q - s(N) - \min\{0, h - s(N)\} = k_q - \min\{s(N), h\} = k_q - T(N). \tag{14}$$

Taking the conditional expectation given all pre-decision information $\mathcal{H}$, including q, shows that the on action has smaller expected cost if and only if $\mathbb{E}[T(N) \mid \mathcal{H}] > k_q$. Resolving equality by the off action gives Eq. (7). Substituting $q = 0$ and $q = 1$ gives thresholds $f + m$ and $f - m$.

No assumption that $f \geq m$ is needed. If $f < m$, the retention threshold is negative and retaining a currently active capability is optimal for the one-step problem even at zero demand. That saving comes from avoiding a larger shutdown charge. The cold threshold remains nonnegative. □

A.3 The information lost by predicting only the mean count

Let $v(\mathcal{H}) = \mathbb{E}[N \mid \mathcal{H}]$ and $\mu(\mathcal{H}) = \mathbb{E}[T(N) \mid \mathcal{H}]$. Since

$$T(N) = \min\{dN, dU, h\}, \tag{15}$$

T is nondecreasing and concave on $[0, \infty)$. Jensen's inequality gives

$$\nu(\mathcal{H}) = T(v(\mathcal{H})) \geq \mu(\mathcal{H}). \tag{16}$$

A controller using the true conditional mean count and applying the deterministic recourse calculation can therefore err only toward the on action relative to the Bayes one-step rule. Its conditional excess cost is

$$(k_q - \mu)\,\mathbf{1}\{\mu \leq k_q < \nu\} \leq \nu - \mu. \tag{17}$$

This is an information limitation of the prediction target, not an estimation error in v. If conditional demand lies entirely on one affine portion of T, the Jensen gap is zero. If its support crosses a clipping threshold, the gap may be positive. Improving the accuracy of v does not recover the missing information about that conditional distribution.

For the example in the main text, $h = 6$ and $k_0 = 5$. Under $N = 5.5$ almost surely, the expected costs of early activation and waiting are 5 and 5.5. Under $N \in \{0, 11\}$ with equal probabilities, their expected costs are 5 and 3. Both distributions have conditional mean 5.5. The mean-count rule activates in both and incurs excess expected cost 2 in the second. A procedure receiving only that conditional mean cannot be Bayes-optimal for both distributions.

A.4 Proof of Proposition 3

Proof. Let $a^* = \mathbf{1}\{\mu > k_q\}$ and $\hat{a} = \mathbf{1}\{\hat{\mu} > k_q\}$. Conditional on $\mathcal{H}$, Eq. (14) gives expected branch difference $k_q - \mu$. If $a^* = \hat{a}$, excess cost is zero. Otherwise it equals $|\mu - k_q|$. In a disagreement, k_q lies between μ and $\hat{\mu}$, possibly at an endpoint, so $|\mu - k_q| \leq |\hat{\mu} - \mu|$. This proves Eq. (8).

For the risk statement, regard the fitted predictor as fixed, or condition on training data independent of the evaluation pair. Assume its squared risk is finite. Conditional expectation gives the orthogonality identity

$$\begin{aligned}\mathbb{E}[(\hat{\mu} - T)^2] &= \mathbb{E}[(\hat{\mu} - \mu)^2] + \mathbb{E}[(\mu - T)^2] \\ &\quad + 2\mathbb{E}[(\hat{\mu} - \mu)\mathbb{E}[\mu - T \mid \mathcal{H}]] \\ &= \mathbb{E}[(\hat{\mu} - \mu)^2] + \mathbb{E}[(\mu - T)^2].\end{aligned}$$

The second term is the Bayes squared risk over all measurable predictors of $\mathcal{H}$. Averaging Eq. (8) and applying Cauchy–Schwarz therefore gives

$$\mathbb{E}[\text{excess one-step cost}] \leq \mathbb{E}[|\hat{\mu} - \mu|] \leq \sqrt{\mathbb{E}[(\hat{\mu} - \mu)^2]} = \sqrt{\mathcal{R}(\hat{\mu}) - \mathcal{R}(\mu)}. \tag{18}$$

This identifies a calibrated target and loss. It does not assert that a finite fitted model has small excess risk, that the chosen context is sufficient, or that summing one-step losses compares different state trajectories. □

A.5 CONTINUATION VALUES IN A FINITE HORIZON

The preceding target is one-step. A finite-horizon optimal controller must also value the final physical state. Suppose the demand process is exogenous, $\mathcal{H}_{t+1}$ is determined by $\mathcal{H}_t, N_t$, and $V_{t+1}(r, \mathcal{H}_{t+1})$ is the optimal future cost for next physical state $r \in \{0, 1\}$. Let

$$D_t(N_t) = V_{t+1}(1, \mathcal{H}_{t+1}) - V_{t+1}(0, \mathcal{H}_{t+1}). \tag{19}$$

With a finite horizon and integrable demand, these values are finite. There is no terminal shutdown charge, so $V_{T+1} = 0$.

For early activation, add $V_{t+1}(1, \mathcal{H}_{t+1})$ to Eq. (12). For waiting, the remote and emergency-local branches respectively add $V_{t+1}(0, \mathcal{H}_{t+1})$ and $V_{t+1}(1, \mathcal{H}_{t+1})$. Subtracting the optimized branches gives

$$C_{1,t}^{\text{Bellman}} - C_{0,t}^{\text{Bellman}} = k_q - \min\{s(N_t) - D_t(N_t), h\}. \tag{20}$$

Thus the finite-horizon activation test replaces $\mathbb{E}[T(N_t) \mid \mathcal{H}_t]$ by

$$\mathbb{E}[\min\{s(N_t) - D_t(N_t), h\} \mid \mathcal{H}_t]. \tag{21}$$

The terminal epoch reduces to Theorem 2. At zero surcharge and a cold state, the clipped continuation-adjusted value still cannot exceed $h = k_0$, so early activation is unnecessary even with optimal future control. Warm retention may depend strongly on the continuation difference. The implemented value regressor uses the one-step target and does not estimate this Bellman quantity. The full-horizon two-state reference in the replay uses the complete realized sequence and is reported separately.

B STATE-CONDITIONED BOUNDS AND SUPPORTING PROPERTIES

The distinction between a forecast optimization problem and an online trajectory is essential when switching couples epochs. We state guarantees at the level supported by the model and implementation. Proofs and counterexamples appear in Appendix C.

Proposition 4 (Feasible execution)**.** *For any nonnegative demand sequence and any planned placements, the common execution rule returns nonnegative y_t, z_t, a physical placement $q_t \geq p_t$, and assignments satisfying Eq. (2). This conclusion holds for all reported policies, not only* SAGE-R.

The property follows from residual-capacity updates and explicit rejection. It does not show that the planner found a good placement. To assess decision quality, fix the actual previous state $s = q_{t-1}$ and let $V_t(n_t, s)$ be the minimum one-epoch cost with demand known in advance, reserve zero, movement weight one, and a single direct transition. Let b_t be a valid lower bound on its LP relaxation. The compact mixed-integer comparison instead optimizes an entire realized sequence from the common initial state.

Theorem 5 (State-conditioned posterior bound)**.** *Every feasible execution of Eq. (2) satisfies*

$$0 \leq C_t - V_t(n_t, q_{t-1}) \leq C_t - b_t. \tag{22}$$

The bound is conditional on the previous state actually reached. Summing these bounds does not produce regret against a different policy or the full-horizon optimum.

The mathematical bound applies to the executed greedy controller and the MPC baselines without an assumed approximation factor. For the reported medium instances, b_t is formed from LP dual multipliers and finite variable bounds. Thus a forecast error, imperfect search, and post-observation repair can all contribute to $C_t - b_t$. The bound measures their combined one-step consequence, not their individual causal effects.

Proposition 6 (Descent under the implemented score)**.** *Let $A_t(x)$ be the fixed-plus-assignment cost computed by the deterministic reserved assignment rule for a fixed forecast. If the planner retains $s = q_{t-1}$ as a candidate and returns p with $A_t(p) + \lambda M(p, s) \leq A_t(s)$, then*

$$\lambda M(p, s) \leq A_t(s) - A_t(p). \tag{23}$$

The local planner has this property. For an additive-δ optimizer of the mathematical program, the corresponding right-hand side is increased by δ.

This concerns the computed forecast score before repair. It is not a bound on the realized movement cost, and A_t need not be the optimum assignment value for a placement. The point of the acceptance test is more limited. A local change cannot increase the very surrogate used to justify that change.

Uniform reserve and joint excess. Suppose a nonnegative tentative assignment w, supported on open compatible bundles of a fixed placement, exceeds capacity by $E_{ik} = [\sum_a d_a^k w_{ai} - U_i^k x_i]_+$. Its minimum removable mass is the value of

$$\min_{0 \le v \le w} \sum_{a,i} v_{ai} \quad \text{subject to} \quad \sum_a d_a^k v_{ai} \ge E_{ik} \quad \forall i, k. \tag{24}$$

The retained assignment is feasible. Removed mass still needs a compatible destination or rejection. Resource-wise excesses cannot be replaced by their minimum. With resource vectors $(1, 0)$ and $(0, 1)$, loads $(2, 2)$, and capacity $(1, 1)$, two units must be removed. Appendix C.4 proves the joint statement and useful upper and lower bounds.

Why there is no global competitive claim. An optimizer of each predicted epoch need not optimize the predicted trajectory, even with perfect forecasts. Consider three unit-demand epochs and two fully compatible facilities. Facility A has residency cost 1 and toggle cost 0.1. Facility B has residency cost 0 and toggle cost 2. Both have zero connection cost and unit capacity. Myopic optimal planning uses A and pays 3.1, whereas the full-horizon optimum uses B and pays 2. The forecast error is zero. A valid full-trajectory perturbation lemma therefore requires a full-trajectory optimization premise. Appendix C.5 states that lemma separately and does not attribute it to the implemented controller.

C Proofs and Scope of the Guarantees

All costs and resource requirements are finite and nonnegative. The sets of types and bundles and the horizon are finite. Demand is a finite nonnegative vector at each epoch. Rejection has finite per-unit cost. Unless explicitly stated otherwise, assignments are divisible. These assumptions make every decision problem feasible through rejection and give a finite optimum. There is no terminal shutdown charge.

C.1 Proof of Proposition 4

Fix an epoch and its proposed binary placement p. The execution rule initializes $q = p$, $y = 0$, and residual resource vector $u_i = U_i q_i$. For a type a, let r_a be its unassigned demand. At a compatible open bundle i, the algorithm assigns at most

$$\delta = \min \left\{ r_a, \min_{k \mid d_a^k > 0} \frac{u_i^k}{d_a^k} \right\}. \tag{25}$$

An empty inner minimum is interpreted as $+\infty$. Thus zero-resource task types cause no division by zero. Every increment is nonnegative, and the update $u_i^k \leftarrow u_i^k - \delta d_a^k$ preserves $u_i^k \ge 0$ in every dimension. Only entries with $K_{ai} = 1$ are updated.

If the rule opens an inactive compatible bundle, it sets $q_i = 1$ and initializes its residual resource vector to U_i. The same update is applied. A bundle can be opened at most once during this epoch, so the loop terminates after finitely many openings and assignments. Rejection receives all remaining unassigned mass. For every type, accepted increments and rejection therefore sum to its original demand. The invariant

$$u_i^k = U_i^k q_i - \sum_a d_a^k y_{ai} \ge 0 \tag{26}$$

holds at termination. The activation link follows because a closed bundle receives no assignment and each type receives at most its offered mass. Compatibility follows from restricting updates to legal edges, and $q \ge p$ follows from allowing only additional openings during repair. The proof is independent of the forecasting and planning routines. Carrying q forward gives the required previous state for the next epoch. No induction hypothesis that yesterday's assignment fits today's demand is needed. □

C.2 Proof of Theorem 5

Define the complete-information one-epoch problem

$$V(n, s) = \min_{x,y,z} \left\{ f^\top x + M(x, s) + \sum_{a,i} c_{ai} y_{ai} + \sum_a R_a z_a \right\}, \tag{27}$$

subject to demand conservation, compatibility, the activation link $y_{ai} \le n_a x_i$, unreserved resource capacity, binary x, and nonnegative y, z. Let (p, q, y, z) be a feasible execution. Then $(x, y, z) = (q, y, z)$ is feasible for Eq. (27). Weighted Hamming distance obeys the triangle inequality coordinate by coordinate, hence

$$M(q, s) \le M(p, s) + M(q, p). \tag{28}$$

Substituting this feasible point into the minimization gives $V(n, s) \le C$. If $L(n, s)$ is the optimum of the LP relaxation and $b \le L(n, s)$ is any valid dual bound, then $b \le L(n, s) \le V(n, s)$. Combining the two inequalities yields $0 \le C - V(n, s) \le C - b$.

The comparator retains the actual previous state s. Two policies generally reach different previous states and therefore different conditional optima. Replacing these state-dependent comparators by one common dynamic optimum would require an additional trajectory argument. No such replacement is made. □

Construction of the numerical lower bound. Write the LP in the form

$$\min_{0 \le v \le u} c^\top v \quad \text{subject to} \quad Av \le b, \quad Ev = h. \tag{29}$$

All upper bounds can be finite. Binary-placement relaxations and transition auxiliaries lie in $[0, 1]$. Demand conservation implies $y_{ai} \le n_a$ and $z_a \le n_a$. For arbitrary multipliers $\mu \le 0$ and unrestricted ν, every feasible v satisfies

$$c^\top v = (c - A^\top \mu - E^\top \nu)^\top v + \mu^\top A v + \nu^\top E v \tag{30}$$

$$\ge \sum_j u_j \min\{0, c_j - (A^\top \mu)_j - (E^\top \nu)_j\} + \mu^\top b + \nu^\top h. \tag{31}$$

The inequality uses both $\mu \le 0$ and the box bounds. The right-hand side is a valid lower bound even when the multipliers do not satisfy stationarity. The implementation takes solver multipliers, projects inequality multipliers onto the nonpositive orthant, and evaluates Eq. (31). It subtracts 10^{-10} times one plus the sum of absolute evaluated terms as a numerical margin. Solver feasibility and the resulting bounds are checked to the reported tolerance. This margin is a numerical safeguard rather than an interval-arithmetic proof of floating-point rounding.

C.3 Proof of Proposition 6

Because $M(s, s) = 0$, retaining the previous state gives candidate value $A_t(s)$. Choosing a candidate with value no larger than this and accepting only decreasing deletion moves gives

$$A_t(p) + \lambda M(p, s) \le A_t(s). \tag{32}$$

Subtracting $A_t(p)$ proves Eq. (23). The local planner retains this incumbent before considering additional candidates, so the argument applies to its returned placement.

For the separate optimization-residual statement, let $H_t(x)$ be the optimal fixed-plus-assignment value of the mathematical forecast problem at placement x, and assume a returned p satisfies

$$H_t(p) + \lambda M(p, s) \le \min_x \{H_t(x) + \lambda M(x, s)\} + \delta. \tag{33}$$

The previous state is admissible because rejection remains available. Comparison with $x = s$ gives

$$\lambda M(p, s) \le H_t(s) - H_t(p) + \delta. \tag{34}$$

A positive part may be applied to the right-hand side, since the left side is nonnegative. Without the descent test or a bound on δ, no corresponding statement follows for an arbitrary heuristic. Neither inequality controls changes introduced by subsequent repair. □

C.4 Uniform slack and the joint removal problem

Let $\hat{y}$ be a compatible reserved assignment at binary placement x, satisfying $\sum_a d_a^k \hat{y}_{ai} \le (1-\epsilon) U_i^k x_i$. Suppose there exists a compatible nonnegative increment Δy at the same placement, satisfying all required additional demand balances and

$$\sum_a d_a^k \Delta y_{ai} \le \epsilon U_i^k x_i \quad \forall i, k. \tag{35}$$

Adding these inequalities proves feasibility of $\hat{y} + \Delta y$. The existence of Δy is an assumption, not a consequence of the reserve fraction. In particular, if a new type is compatible only with a closed facility, slack at another site cannot serve it.

For a nonnegative tentative assignment w supported on open compatible bundles, define $L_{ik} = \sum_a d_a^k w_{ai}$ and $E_{ik} = [L_{ik} - U_i^k x_i]_+$. For any $v \in [0, w]$, the retained mass $w - v$ is resource-feasible if and only if

$$\sum_a d_a^k v_{ai} \geq L_{ik} - U_i^k x_i \quad \forall i, k. \tag{36}$$

The left side is nonnegative, so this is equivalent to the constraints of Eq. (24). The feasible set is nonempty because $v = w$ is feasible. It is closed and bounded, so a minimizer exists. Compatibility is preserved by subtraction. This proves that the LP value is the minimum mass removed from this tentative assignment. It is not necessarily the minimum globally rerouted mass, because rerouting can alter other facilities' loads.

For a facility i, let $N_i = \sum_a w_{ai}$, and define

$$d_{\min,ik} = \min_{a \,|\, w_{ai}>0,\ d_a^k>0} d_a^k, \qquad d_{\max,ik} = \max_{a \,|\, w_{ai}>0} d_a^k. \tag{37}$$

Dimensions with $E_{ik} = 0$ need not be considered. An empty maximum or sum is zero. If $E_{ik} > 0$, the relevant minimum and maximum exist and are positive. Writing r_i^* for the optimal removed mass at facility i, we obtain

$$\max_{k \,|\, E_{ik}>0} \frac{E_{ik}}{d_{\max,ik}} \;\leq\; r_i^* \;\leq\; \min\left\{ N_i,\ \sum_{k \,|\, E_{ik}>0} \frac{E_{ik}}{d_{\min,ik}} \right\}. \tag{38}$$

For the lower bound, any removed mass r removes at most $d_{\max,ik} r$ in dimension k. For the upper bound, process dimensions sequentially. Whenever an excess remains in dimension k, remove divisible mass from tasks consuming that resource until the excess vanishes. Enough consuming mass exists because the current resource load exceeds capacity. Each removed unit contributes at least $d_{\min,ik}$, so this step removes at most the current excess divided by $d_{\min,ik}$, and hence at most $E_{ik}/d_{\min,ik}$. Removal cannot increase a previously treated excess. Summing these bounds and observing that no more than all mass can be removed proves the result. Distinct facilities have independent removal variables, so the total optimum is $\sum_i r_i^*$. □

C.5 Full-trajectory transfer and a counterexample to myopic consistency

Let $n = (n_1, \ldots, n_T)$ and $\hat{n} = (\hat{n}_1, \ldots, \hat{n}_T)$. All costs, masks, and capacities are the same in the two instances. Put $L = \max_a R_a$. A fixed feasible trajectory for $\hat{n}$ can be converted into one for n without changing any planned or physical placements. When demand decreases, remove assigned or rejected mass of that type. Nonnegative resource consumption and connection costs imply that this cannot increase cost. When demand increases, reject the additional mass at cost at most L per unit. Consequently there exists a repair operator $\mathcal{R}_0$ with

$$C(\mathcal{R}_0(\hat{\pi}, n), n) \leq C(\hat{\pi}, \hat{n}) + L \sum_t \|n_t - \hat{n}_t\|_1. \tag{39}$$

Applying the same construction to an optimal trajectory in each direction gives

$$|\operatorname{OPT}(n) - \operatorname{OPT}(\hat{n})| \leq L \sum_t \|n_t - \hat{n}_t\|_1. \tag{40}$$

This establishes existence of a bounded rejection-based repair. It does not prove that an arbitrary greedy opening-based repair has the same bound, particularly if it changes future physical states.

Lemma 7 (Full-trajectory perturbation transfer)**.** *Suppose the full predicted sequence is available to a planner and its returned trajectory satisfies $C(\hat{\pi}, \hat{n}) \leq \alpha \operatorname{OPT}(\hat{n}) + \Delta$. If execution uses a repair operator satisfying Eq. (39), then*

$$C(\mathcal{R}_0(\hat{\pi}, n), n) \leq \alpha \operatorname{OPT}(n) + \Delta + (\alpha + 1) L \sum_t \|n_t - \hat{n}_t\|_1. \tag{41}$$

Proof. Use the repair inequality, substitute the full-trajectory optimization premise, and apply the Lipschitz inequality to $\mathrm{OPT}(\hat{n})$. These three steps give

$$\begin{aligned} C(\mathcal{R}_0(\hat{\pi}, n), n) &\le C(\hat{\pi}, \hat{n}) + LE \\ &\le \alpha\,\mathrm{OPT}(\hat{n}) + \Delta + LE \\ &\le \alpha\,\mathrm{OPT}(n) + \Delta + (\alpha+1)LE, \end{aligned}$$

where $E = \sum_t \|n_t - \hat{n}_t\|_1$. None of these steps changes an epoch-wise approximation assumption into a trajectory-wise one. □

The two-facility example in Appendix B makes the distinction quantitative. At the first epoch, costs are 1.1 for opening A, 2 for opening B, and at least 100 for rejection. Once A is open, retaining it costs 1 and switching to B costs 2.1. Thus optimal one-step decisions cost 3.1. Over the three-epoch horizon, opening B once costs 2. Any trajectory that activates B pays at least 2, and the best trajectory that never activates B costs 3.1. Therefore the dynamic optimum is 2. Increasing capacities to $1/(1-\epsilon)$ gives the same counterexample with a positive reserve. Perfect forecasts make every forecast-error term zero, so they cannot repair the incorrect myopic comparison.

C.6 Expected expert selection is not a pathwise guarantee

This subsection states a separate generic result to make the probability and state assumptions explicit. It is not used by SAGE-R or any reported experiment. There are M virtual experts with a fixed loss table $\ell_{t,e} \in [0, G]$. Each expert is simulated from its own previous state, all expert losses are revealed after a round, and the loss table is fixed before the master's random choices. Let

$$p_{t,e} = \frac{\exp(-\eta \sum_{s<t} \ell_{s,e}/G)}{\sum_j \exp(-\eta \sum_{s<t} \ell_{s,j}/G)}. \tag{42}$$

The master draws $E_t \sim p_t$ and implements expert E_t. Assume this can be done feasibly at cost at most $\ell_{t,E_t} + S_t$, where $S_t \ge 0$ is the actual cost of synchronizing physical state to that expert. Let $S = \sum_t S_t$.

For $x \in [0,1]$, the exponential potential bound, equivalently Hoeffding's lemma, gives

$$\log \sum_e p_{t,e} \exp(-\eta \ell_{t,e}/G) \le -\frac{\eta}{G} \sum_e p_{t,e} \ell_{t,e} + \frac{\eta^2}{8}. \tag{43}$$

To obtain the constant directly, apply the usual bounded-variable moment bound to the random variable taking values $\ell_{t,e}/G$ with probabilities $p_{t,e}$. Its range has length at most one. For completeness, write $\psi(u) = \log \mathbb{E}[e^{u(X - \mathbb{E}X)}]$ for any such variable. Then $\psi(0) = \psi'(0) = 0$ and $\psi''(u)$ is the variance under an exponential tilt. A variable in an interval of length one has variance at most $1/4$, including under that tilt. Integrating twice gives $\psi(u) \le u^2/8$. Summing the log potential and comparing its terminal value with each individual expert yields

$$\sum_t \sum_e p_{t,e} \ell_{t,e} \le \min_e \sum_t \ell_{t,e} + G\left(\frac{\log M}{\eta} + \frac{\eta T}{8}\right). \tag{44}$$

Conditional expectation of the sampled loss equals the mixture loss. Taking expectations and choosing $\eta = \sqrt{8 \log M/T}$ therefore proves

$$\mathbb{E}[C] \le \min_e C_e + G\sqrt{T \log M/2} + \mathbb{E}[S]. \tag{45}$$

For a high-probability version, the difference $\ell_{t,E_t} - \sum_e p_{t,e} \ell_{t,e}$ is a martingale difference whose conditional range has length at most G. The conditional bounded-variable moment bound and exponential Markov inequality give, with probability at least $1 - \delta$,

$$C \le \min_e C_e + G\sqrt{T \log M/2} + G\sqrt{T \log(1/\delta)/2} + S. \tag{46}$$

No competitive guarantee follows unless the expert comparator and synchronization cost are themselves controlled. In particular, S may be linear in T. No fallback with an unproved competitive ratio is presumed here.

For comparison, with expert losses $(0,0)$ and $(1,1)$, a sampled master can choose the second expert twice and incur cost 2. The event has probability $\frac{1}{2}/(1+e^{\eta}) > 0$ under full-information updates. For $\eta = \sqrt{\log 2/4}$ this is about 0.1987. A deterministic bound of $\sqrt{4 \log 2} < 2$ is therefore false. The expected and high-probability statements above do not make that pathwise claim.

C.7 Compatibility and metric charging

Let a physical metric place a client a at coordinate zero, an offline facility o at one, and an already-open facility j at two. Set $K_{ao} = 1$ and $K_{aj} = 0$. Replacing an inadmissible connection by infinite cost gives $d_K(a, j) = +\infty$, although $d(a, o) + d(o, j) = 2$. Thus a metric charging argument that treats any facility near o as a possible connection for a cannot be used after simply deleting infeasible edges. This is a failure of that proof step, not a proof that the problem lies outside established nonmetric facility location. It motivates keeping compatibility explicit in all planning and comparison procedures.

D Matched calibration, optimization diagnostics, and call replay

D.1 Fit, calibration, and test separation

The primary medium study uses the same type and facility generator as Appendix G, with 12 types and 24 ordinary plus three fallback bundles. Its training stream has 120 epochs of 500 units, generated with seed offset 1,000. The first 96 epochs fit the shared ridge coefficients and scale. The last 24 epochs select forecast configurations. The independent test stream has 24 epochs and seed offset 2,000.

The EMA and ridge implementations are identical across local and MPC arms. Calibration histories begin at the fitted training prior, then receive actual calibration observations. Test forecast history is initialized by the calibration observations. Physical state is reset to all closed at the beginning of every calibration and test rollout. Predictions do not receive current or future test demand, and they are not rescaled to its known experimental total.

The fixed nine-point grid in Eq. (4) contains both pure EMA and pure ridge. Count selection minimizes one-step squared count error on calibration. Local and MPC cost selection each minimize the complete calibration rollout objective, including all residency, toggle, rejection, and repair charges. Selection is global to the test run, not changed after observing test outcomes. In ties the lowest grid index is retained. The supplement includes outcomes for every calibration candidate, not only the selected forecast.

There are 18 calibration rollouts per condition, one for every forecast candidate through each solver, giving 216 calibration rollouts across 12 conditions. Five frozen forecast arms are then crossed with two solvers, giving 120 test rollouts. The arms are EMA, count ridge, count-selected grid, local-cost-selected grid, and MPC-cost-selected grid. Runtime is separated into fitting, calibration, and test evaluation, including the additional calibration cost of the cost-selected forecasts.

The broader medium suite in Appendix F comprises 84 runs of seven policies, namely SAGE-R, local EMA, local last-value, local ridge, MPC EMA, MPC last-value, and MPC ridge. Local last-value results are included in the supplement and omitted from the six-column family table for space. This suite uses a larger fitting sample and a combined node and clock budget. Its results are therefore reported separately from the primary calibration study.

D.2 Solver status and finite budgets

The primary MPC solves use horizon three, reserve 0.07, movement weight 0.95, a 64-node branch-and-bound cap, and relative gap target 10^{-7}. The horizon shortens at the end of each rollout. Solves start without a warm-start incumbent and have no wall-clock cap. Optimization quality is assessed from termination status, incumbent objectives, dual bounds, and relative and absolute gaps. Node counts and elapsed time characterize the computational cost. When no feasible incumbent is available, the common local planner supplies the action, and the outcome is classified as fallback rather than optimal termination.

All 1,440 primary test calls returned optimal status. The maximum reported relative gap was 2.28×10^{-14}, and no fallback occurred. These are achieved solver outcomes rather than requested tolerances. Numerical feasibility and optimality tolerances remain distinct from a real-arithmetic proof.

Table 3. Common-input solver replay on 96 fixed states and three forecasts. Clock-capped inputs have two timed repetitions. All limited terminations arose from the time cap. All calls returned incumbents and none used the local fallback. Gaps are the solver's reported relative gaps.

Budget	Forecast	Calls	Optimal	Limited	Max gap
64 nodes	EMA	96	96	0	$< 10^{-10}$
64 nodes	Count ridge	96	96	0	$< 10^{-10}$
64 nodes	Cost-selected MPC	96	96	0	$< 10^{-10}$
64 nodes, 0.5 s	EMA	192	192	0	$< 10^{-10}$
64 nodes, 0.5 s	Count ridge	192	192	0	$< 10^{-10}$
64 nodes, 0.5 s	Cost-selected MPC	192	192	0	$< 10^{-10}$
64 nodes, 0.05 s	EMA	192	188	4	1.29%
64 nodes, 0.05 s	Count ridge	192	176	16	80.88%
64 nodes, 0.05 s	Cost-selected MPC	192	188	4	5.25%

The budget replay fixes the first eight actual states of the EMA-local trajectory for each of 12 conditions. At each state, EMA, ridge, and MPC-calibrated forecasts are held fixed and passed to the same optimization model. The node-only run is performed once, while the 0.5-second and 0.05-second versions are each repeated twice. These repeated calls are measurements of the same optimization problem, not additional independent demand instances. The 1,440 calls comprise 288 node-only, 576 half-second, and 576 short-limit solves.

The short-limit ridge arm has both more interrupted solves and a larger worst gap. A difference under a common wall-clock cap therefore cannot be attributed to the forecast alone when incumbent quality also differs. The uncapped-clock primary comparison avoids that ambiguity on the evaluated inputs. It does not establish that all larger MPC instances will solve within 64 nodes or within a practical clock deadline.

D.3 Fixed-state recourse counterfactual

For each of the same 96 states, a local EMA plan p is compared with $p \wedge s$ at surcharges $\gamma \in \{0, 20, 100\}$. The one-epoch recourse solver requires $q \geq p$, uses realized demand, reserve zero, and movement weight one. It includes finite capacities and rejection. The deferred problem uses the same previous state, costs, and realized demand. It is not a rollout from the deferred policy's own earlier decisions.

To express recourse in the common optimizer, increase each final residency coefficient by γ, constrain the first placement to contain p, and subtract $\gamma \sum_i p_i$ afterward. Add the already-incurred planned transition $M(p, s)$. This yields precisely the objective in Eq. (2). Both arms use a 100,000-node limit without a clock cap. All 576 solves returned an incumbent and gaps below 8.0×10^{-15}.

Surcharge	Pairs	Mean difference	Minimum	Maximum
0	96	−3.30	−114.96	0.00
20	96	2.62	−74.96	40.00
100	96	22.77	−20.69	200.00

Differences are deferred minus early modeled cost, not percentages. There are 42 newly planned activations across these states. Negative differences reflect activations that optimal delayed recourse can omit. No paired difference exceeds the identity's upper bound $\gamma|p \setminus s|$ after numerical tolerance.

D.4 Public call projection and separable replay

The replay uses 40 public API-Bank dialogues from multi-API task families in the Level-1 description partition. Each selected family contributes its lowest-indexed available example. The subset was

fixed before placement evaluation and is smaller than the complete 314-dialogue benchmark. The source selection is documented in the supplement.

Demand follows the explicit API events in each dialogue, in their annotated order. Task labels and natural-language descriptions of completed actions are not treated as API events. For example, a dialogue labeled with a stock-query task may contain only an annotated bank-account opening. One selected dialogue has no explicit API events and contributes a zero-demand epoch. The resulting collection contains 104 calls, 26 API identities, and 38 distinct sequences.

Only API identities and their order enter the demand model. Credentials, patient information, argument values, and dialogue text are excluded from the shared projection. The source consists of public benchmark annotations, not observed service traffic. Attribution and licensing accompany the data in the supplement.

Each dialogue is an independent reset episode. Consecutive blocks of at most two calls form decision epochs, with a final shorter block when needed. Every call contributes one count to its API type. There is no interpolation, resampling, or invented timestamp within this projection. Different dialogues are not concatenated into a supposed production time series. The within-dialogue order is preserved, but the experiment does not execute tools, infer workflow dependencies, or evaluate whether an annotated answer is correct.

The separable local–remote model has $b = 1$, $c = 0.2$, and one optional local capability per API identity. Each type has its own capacity and compatible remote destination. The remote option is always available and is not a permission exception. The fixed mapping grid is

$$U \in \{1, 2\}, \quad f \in \{0.05, 0.2, 0.5\}, \quad m \in \{0.05, 0.3, 0.8\}, \quad \gamma \in \{0, 0.5, 2\}. \tag{47}$$

All 54 mappings are evaluated. They are counterfactual prices and capacities, not estimates of API runtime, memory demand, or transfer cost. Shared capacity and cross-tool state conservation are deliberately absent so that Theorem 2 applies to each capability.

Five-fold splits group identical projected sequences together before assigning folds with a fixed random seed. This prevents duplicate sequence aliases from appearing on both sides of a fold. The API vocabulary is a known capability inventory and is common to all folds. Counts and target means are fitted using only training dialogues.

Both regressors use the last previously observed API identity, or a start marker, as their context. If N_h training epochs have context h, the fitted conditional estimate for target Y is

$$\widehat{\mathbb{E}}[Y \mid h] = \frac{\sum_{j \mid h_j = h} Y_j + \bar{Y}}{N_h + 1}, \tag{48}$$

where $\bar{Y}$ is the overall training mean. Unseen contexts use $\bar{Y}$. The smoothing weight is fixed at one for both regressors and is not selected on test performance. Count regression uses $Y = n$ and applies T to its output. Value regression uses $Y = T(n)$ directly. They share the same previous-state threshold k_q and the same optimal immediate repair decision. Their restricted context need not equal the full conditional information in Theorem 2, so neither fitted regressor is asserted to be Bayes-optimal.

Every dialogue is held out once in each mapping. The reference arms use the previous observed block, a no-advance rule, and a clairvoyant one-step decision. A separate full-horizon reference optimizes each API's binary state trajectory using a two-state backward dynamic program on the actual counts. All starts are closed, and no terminal shutdown cost is added. The clairvoyant one-step arm need not equal the full-horizon optimum. No achieved controller cost is below the latter reference within numerical tolerance.

Matched decision diagnostics and references. At each test context, both fitted targets are evaluated from the count controller's previous state. Since both regressors use the same positive empirical weights, Jensen's inequality also holds for their fitted values. The count-transformed estimate is never below the direct clipped estimate. All 86 observed disagreements therefore remove an on decision rather than create one. The 20 zero-surcharge disagreements concern warm-state retention. There are no cold-state disagreements at zero surcharge. Evaluating the two changed actions on realized demand gives 82 lower and four higher one-step costs. These 86 events are drawn from

87,048 repeated API-state coordinates across the fixed mappings, not from that many independent observations.

For completeness, the aggregate modeled costs per offered call, pooled over all fixed mappings, are

Policy	Cost per offered call
Count regression	0.907470
Clipped-value regression	**0.905849**
Previous block	0.963275
No advance activation	0.912153
Clairvoyant one-step reference	0.763462
Full-horizon two-state reference	0.759348

Bold marks the lowest displayed non-clairvoyant cost. The two reference arms use information unavailable to the forecasting policies. These pooled unit costs are not the same statistic as the mean of mapping-wise percentage gains in Table 2. The remaining gap also shows why the finite-context regressors should not be described as attaining the conditional or full-horizon optimum.

The 10,800 replay rollouts are $54 \times 40 \times 5$ policy evaluations of the same small input collection. They are not 10,800 independent workloads. The six positive mapping effects and 48 ties in Table 2 are reported without treating those mapping repetitions as independent samples for a significance test. Public data supply call identity and sequence only. The study does not locate actual deployments in the chosen cost–capacity space.

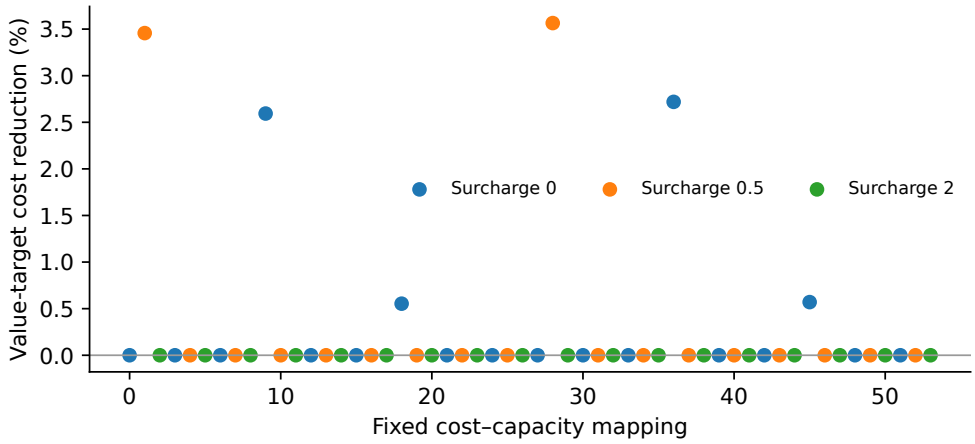

Figure 4. All fixed annotated-call replay mappings. Positive values favor clipped-value regression over count regression. Six mappings change the aggregate held-out cost and 48 are ties. No mapping is selected as representative after observing its result.

E HISTORICAL PORTFOLIO AND MATCHED ABLATIONS

Portfolio selection. SAGE-R evaluates $4 \times 3 \times 2 = 24$ configurations. The reserve grid is $\{0.03, 0.07, 0.11\}$ and the movement grid is $\{0.45, 0.95\}$. Each configuration produces a planned placement from the same current physical state. Its score is the mean of its realized one-epoch costs on the last four epochs, each evaluated from the state actually present at that epoch. The selected configuration is fixed before current demand is observed. Afterward, the execution cost of each candidate is computed to update its score. These counterfactual one-step losses do not represent independent stateful expert trajectories.

The cold-start configuration is EMA with $\epsilon = 0.07$ and $\lambda = 0.95$. The normalization used in historical scores divides cost by $\max\{1, \mathbf{1}^\top n_t\}$. All component-removal controls use the same 24 slots, which need not be distinct configurations and scoring rule. Disabling a component replaces

Algorithm 2 SAGE-R with common physical-state accounting

Input compatible bundles, training stream, configuration set Θ
$q_0 \leftarrow 0$ and $\mathcal{L} \leftarrow [\,]$
for $t = 1, \ldots, T$ **do**
  for $\theta = (g, \epsilon, \lambda) \in \Theta$ **do**
    $\hat{n}_t^\theta \leftarrow g(n_{<t})$
    $p_t^\theta \leftarrow \mathrm{LocalPlan}(\hat{n}_t^\theta, q_{t-1}, \epsilon, \lambda)$
  end for
  $\theta_t \leftarrow \arg\min_\theta \mathrm{MeanLastFour}(\mathcal{L}_\theta)$
  Observe n_t and execute $p_t^{\theta_t}$ under Eq. (2)
  Record q_t and compute all candidate costs from q_{t-1} on n_t
  Append normalized candidate costs to $\mathcal{L}$
end for

its values by a common value without reallocating that budget. We also report single-configuration controls, so the cost of the portfolio itself is visible.

F BROAD GENERATED SUITES AND EARLIER MATCHED COMPARISON

The empirical question is whether forecasting improves placement under a common execution model, not whether a large simulator run constitutes deployment evidence. All results below were generated with the controller and accounting in Sections 2 and 3. These studies use the same physical-state accounting as the main experiments. The medium comparison here uses the earlier training protocol and clock-limited optimizer. It is distinct from the fit/calibration/test comparison in Section 5.

Data and protocol. Four synthetic scenario families use coarse capability categories inspired by API-Bank, ToolBench, MCPToolBench++, and AppWorld. Their labels do not mean that benchmark dialogues or tool calls were loaded. Capacities, permissions, regions, locality costs, and nonstationary counts are generated. Training and evaluation use separate seeds. The main CPU study has 12 paired scenario-seed conditions, 60 epochs per run, 1,000 demand units per epoch, and 64 ordinary bundles plus three finite-capacity fallback bundles. Its 120 policy runs evaluate ten distinct controllers. Repeated demand mass and internal candidate evaluations are not treated as independent observations.

The medium study uses the same four families and three seeds, with 24 epochs, 500 units per epoch, 12 types, and 24 ordinary bundles plus three fallback bundles. Seven policies give 84 runs. Each MPC call uses horizon three, a 64-node limit, a 0.5-second time limit, and relative gap target 10^{-7}. The local baselines in this study receive the same forecasts and execution rules. The compact study solves eight six-epoch full-horizon instances with eight types and 15 bundles. All eight horizon solves terminate with reported primal–dual gaps below 10^{-7}.

Metrics and uncertainty. Tables report total realized objective divided by offered demand, including rejection. Connection costs are dimensionless model quantities, not milliseconds. Repair fractions count all mass assigned to a bundle opened after demand was observed. Repair-attributed cost includes that mass's connection cost and the incremental residency, opening, and surcharge costs. Rejection is separate. Runtime includes planning, counterfactual scoring, and execution, with posterior numerical-bound computation included in the medium study. Training time is reported separately in the supplement.

For paired policies A, B, percentage improvement is $100(1 - C_A/C_B)$ in each scenario-seed condition. Reported means weight the four families equally. The 95% bootstrap intervals use 20,000 resamples of seeds within each family, holding the family set fixed. With only three or five seeds per family, these intervals summarize the evaluated conditions rather than uncertainty over a population of agent deployments. The supplement gives per-seed results, including losses.

Equal-budget ablations. Table 4 removes one component at a time while retaining the score, history window, and 24 candidate evaluations. SAGE-R reduces cost relative to the no-forecast control by 2.46% with interval $[0.74, 4.31]$. Removing reserve or switching regularization has a smaller and less consistent effect. Neither comparison has an interval excluding zero. Against the

Table 4. Main CPU suite. Component removals retain 24-slot selection. EMA-Fixed is the separate single-configuration reference. Entries are modeled cost per offered demand unit, averaged over three seeds. Bold marks the lowest displayed family mean.

Policy	API	Tool	MCP	App
SAGE-R	1.660	2.076	1.758	1.918
EMA-Fixed	1.669	2.073	**1.751**	1.938
NoForecast	1.692	2.165	1.796	1.962
NoReserve	**1.653**	2.118	1.778	1.939
NoSwitchPenalty	1.735	**2.072**	1.771	**1.912**

Table 5. Matched medium instances. All methods share the actual execution rule. MPC uses the same optimization limits across forecasts. Entries are modeled cost per offered demand unit.

Policy	API	Tool	MCP	App
SAGE-R	1.951	2.349	2.219	2.066
EMA-Fixed	1.952	2.425	2.221	2.083
Ridge-Fixed	1.945	2.533	2.224	2.116
MPC-Last	1.742	2.079	1.821	1.864
MPC-EMA	1.736	2.071	1.801	1.832
MPC-Ridge	**1.727**	**2.045**	**1.785**	**1.809**

single-configuration EMA control, SAGE-R improves by only 0.21% with interval $[-0.49, 0.88]$. Thus this study supports using temporal demand information relative to a training-prior control, but does not establish that portfolio search reliably improves on fixed EMA. Fixed-parameter ablations are reported separately in Appendix H.

Forecast quality and downstream optimization. The trained ridge model reduces one-step count error relative to EMA by 15.42% with interval $[14.50, 16.44]$ on the medium suite. Under local search, however, its cost improvement is -1.51%, meaning that cost increases. Under MPC, its cost improves by 0.95% with interval $[0.43, 1.52]$, and it wins 10 of 12 pairs. The larger difference in Table 5 is the optimizer effect. MPC-EMA improves by 13.18% over local EMA on these instances. We therefore do not attribute the MPC comparison to forecasting alone. Figure 5 isolates the within-controller forecast comparison. Search quality and the predictor's effect on discrete placements must both be measured.

A100-calibrated simulation. The calibration covers one A100-SXM4-40GB, a dense-matrix kernel rate of 250.6 TFLOP/s, and host-to-device and device-to-host rates of 11.5 and 12.3 GiB/s. The 58.7k/s attention-kernel output rate is not a measure of full autoregressive decoding and does not enter the capacity mapping. Appendix G.5 gives the hand-chosen mapping from the used measurements to modeled bundle capacities and costs. No new GPU kernel measurements or live model-serving runs were conducted for this revision.

Table 6 reports the calibrated study with 20 paired family-seed conditions, 125 epochs, 8,000 units per epoch, and 128 ordinary plus three fallback bundles. Its four policies account for 80 runs. With common fallback persistence and two-stage transition charges, SAGE-R improves cost over EMA by 1.55% with interval $[0.64, 2.69]$, winning 17 of 20 pairs. These are new controller results, not a reinterpretation of the previous implementation. Emergency openings serve 0.407% of offered mass for SAGE-R versus 0.293% for EMA. Corresponding rejection rates are 1.239% and 1.251%. Zero incompatibility holds for both by the shared execution rule and is not a comparative advantage. The result is confined to the calibrated simulator.

Repair and persistence. Table 7 reports every valid CPU policy under the same stage attribution. Repair serves 0.521% of offered mass for SAGE-R and accounts for 0.757% of its objective. The respective values for fixed EMA are 0.514% and 0.773%. The two controllers therefore have similar repair exposure on this suite. Removing reserve increases emergency-served mass in both the fixed and matched-budget comparisons. Regular retirement is exercised rather than merely permitted

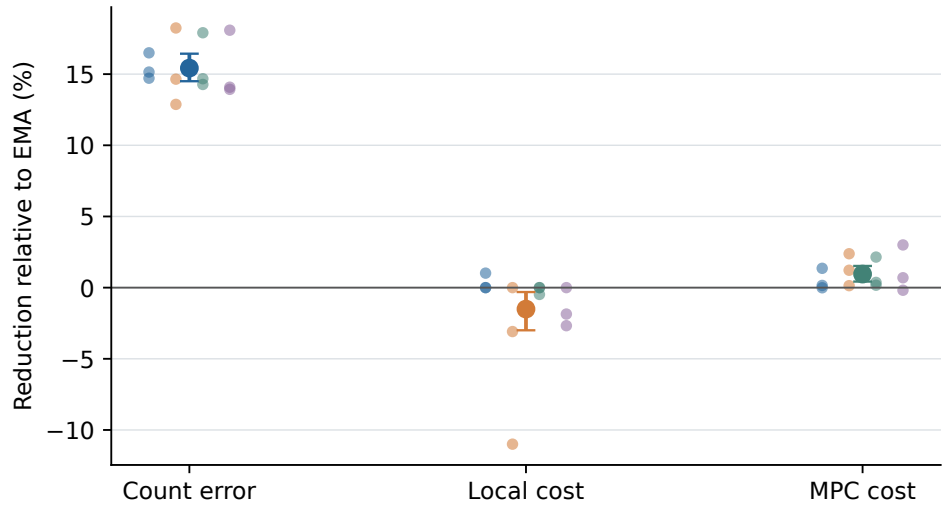

Figure 5. Ridge versus EMA on the same medium instances. Positive values favor ridge. The forecast-error reduction is identical for the local and MPC comparisons. Small points show all 12 paired conditions. Large markers and bars show the mean and within-family bootstrap interval. Better count prediction does not imply lower local-search cost.

Table 6. A100-calibrated simulations with common persistence and paid repair. Entries are modeled cost per offered demand unit, averaged over five seeds. These are CPU simulations using a measured calibration, not model-serving measurements.

Policy	API	Tool	MCP	App
SAGE-R	**1.892**	5.728	**7.072**	3.736
Last-Fixed	1.959	5.824	7.244	3.792
EMA-Fixed	1.962	5.746	7.147	3.749
Ridge-Fixed	1.949	**5.695**	7.090	**3.723**

by the formulation. Across the 12 runs, SAGE-R closes 546 bundles and fixed EMA closes 373. These counts establish that the revised study includes expansion and consolidation. They do not measure memory-object transfers or prove that a particular closure was optimal. Repair shares are cost attributions, not causal estimates of how costs would change if repair were unavailable.

Sensitivity and bounds. We independently vary capacity, toggle cost, forecast mixing, compatibility sparsity, family-shift strength, demand concentration, and emergency activation surcharge in 540 short stress runs. Gains change sign across settings and families. In particular, stronger family shifts do not uniformly favor SAGE-R. The complete sweeps appear in Appendix G.6. Compact solver comparisons and state-conditioned numerical bounds are reported in Appendix H. The former compare full trajectories. The latter diagnose one-step losses from each policy's actual state. Neither is used as a substitute for a missing deployment experiment.

G EXPERIMENTAL SPECIFICATION

G.1 GENERATED SCENARIO CONSTRUCTION

The four generated placement families use capability labels inspired by the cited tool-use benchmarks. They do not use benchmark dialogues or tool-call sequences. The separate API-Bank study in Appendix D.4 uses annotated call identities. Benchmark sizes are not treated as observed serving demand. All parameters below are simulation choices rather than estimates fitted to the benchmarks.

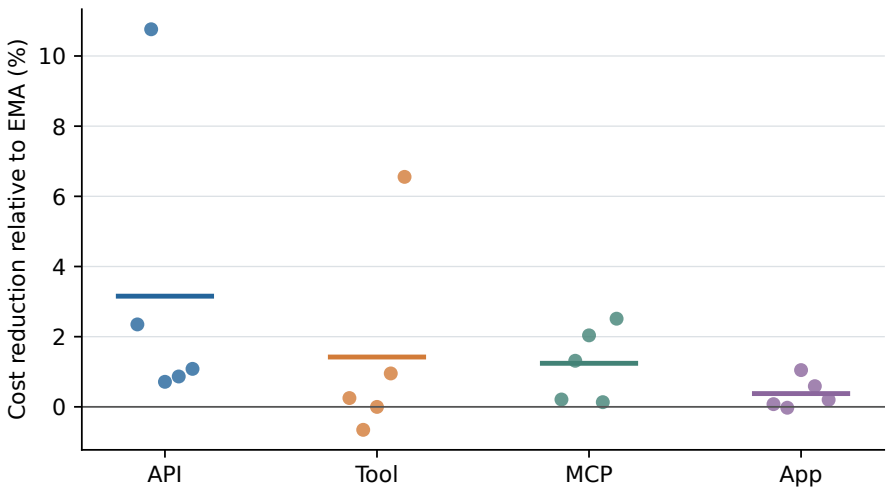

Figure 6. All 20 calibrated-simulation comparisons of SAGE-R with fixed EMA. Positive values favor SAGE-R. Horizontal segments show family means. The between-seed range, including losses, is retained rather than represented by the aggregate gain alone.

Table 7. CPU repair accounting across all ten policies. Mass and rejection percentages divide by offered mass. Repair cost divides by total objective. Closures are planned retirements over all runs. Tiny nonzero rejection rates are not rounded to zero.

Policy	Repair mass	Repair cost	Rejection	Closures
SAGE-R	0.521%	0.757%	<0.001%	546
Last-Fixed	0.561%	0.795%	<0.001%	686
EMA-Fixed	0.514%	0.773%	<0.001%	373
Ridge-Fixed	0.411%	0.677%	<0.001%	153
Prior-Fixed	0.462%	0.790%	<0.001%	135
EMA-NoReserve	1.031%	1.501%	<0.001%	365
EMA-NoSwitchPenalty	0.624%	0.870%	<0.001%	699
NoForecast	0.481%	0.766%	<0.001%	154
NoReserve	0.978%	1.427%	<0.001%	416
NoSwitchPenalty	0.825%	1.157%	<0.001%	779

The default numbers of types are 24, 36, 30, and 27 for the API, Tool, MCP, and App families. Types cycle through the corresponding capability names, with variants having different resource needs, permission levels, and regions. Permission levels lie in $\{0, 1, 2, 3\}$. Resource demands depend on the variant and a bounded random factor. File, database, code, finance, and health labels receive specified resource multipliers. A stateful flag changes the modeled locality cost. It does not introduce a memory-object identifier or an ownership constraint.

Candidate ordinary bundles are specialists, multi-family bundles, or generalists. Specialists have one or two families and base capacities $(42, 32)$. Multi-family bundles have two to four families and base capacities $(68 + 6s, 50 + 5s)$ for s supported families. Generalists have base capacities $(118 + 8s, 90 + 7s)$. Each ordinary capacity receives a multiplicative draw in $[0.86, 1.14]$ and is divided by the family-specific stress factor. Three additional fallback bundles have capacity $(420, 340)$ each and recurring costs 95, 110, and 125. These capacities are finite. Fallback bundles are available during planning and repair and follow the same carry-and-close rule as ordinary bundles.

A task can use a bundle only if its family is supported, its required permission does not exceed the bundle's level, and the region predicate holds. The six labels are us-east, us-west, eu, asia, edge, and gov. Government-labeled tasks are confined to the government-labeled region. Higher-permission eu-labeled tasks are restricted to eu or gov. These predicates are a toy policy model and not a legal-compliance claim. Connection cost is derived from the absolute difference of the integer region indices, multiplicative locality factors, capability breadth, and a reliability divisor. It has no calibrated time unit and does not model Internet distances.

Demand probabilities have a long-tailed base over type variants. Specified family weights grow after 35% of the horizon, a rotating family receives a modest periodic burst multiplier, and higher-permission types receive an additional late-horizon factor. An epoch probability vector is sampled from a Dirichlet distribution with concentration $100p + 0.35\mathbf{1}$, followed by a multinomial count draw. Main-study epoch totals are fixed. Training uses at least 60,000 demand units, a seed offset of 1,000, and a separate stream identifier. Evaluation uses an offset of 2,000 and another identifier. Type and facility construction use their own stream identifiers. Although random draws are disjoint, training and evaluation share the generator. This is not an external generalization test.

G.2 Forecast model

Let μ_a be the training mean for type a and $s = \max(1, |\mathcal{A}|^{-1} \sum_a \mu_a)$. The pooled ridge design has nine features,

$$\phi_{a,t} = \left(1, \frac{n_{a,t-1}}{s}, \ldots, \frac{n_{a,t-6}}{s}, \frac{\mu_a}{s}, \frac{n_{a,t-1} - n_{a,t-3}}{s}\right). \tag{49}$$

A common coefficient vector is fitted to all type-time observations after the six-epoch training prefix by minimizing

$$\sum_{a,t} \left(\phi_{a,t}^\top \beta - \frac{n_{a,t}}{s}\right)^2 + 0.1 \sum_{j=2}^{9} \beta_j^2 + 10^{-8} \beta_1^2. \tag{50}$$

Predictions are $\hat{n}_{a,t} = s[\phi_{a,t}^\top \beta]_+$. The coefficients and the scaling statistics remain frozen during evaluation. Features use only the observed prefix, initialized by the final six training epochs. Multi-step forecasts are recursive. No current or future evaluation count enters a forecast or configuration selection.

The EMA is initialized at the oldest of the six available lags and updated over the next five using $v \leftarrow 0.65n + 0.35v$. Last-value prediction returns the most recent observed vector. The prior returns μ. Forecast vectors are nonnegative but are not rescaled to the actual current epoch total. All fixed local and MPC forecasting comparisons use these same routines.

G.3 Local search and matched controls

The construction of a candidate open set visits types in increasing order of their number of admissible bundles. Ties use type index. Existing compatible capacity is considered in increasing modeled connection cost. When a cover construction needs a new bundle, it selects among dormant compatible candidates using a per-unit score consisting of connection cost and the additional recurring and weighted toggle costs divided by assignable mass. Candidates more expensive than rejection are not opened. This construction is a heuristic. It need not minimize Eq. (3).

The previous placement, a cover initialized from it, and a cover initialized from the empty placement are scored using the same deterministic reserved assignment rule. The best is retained. At most four active bundles, ordered by utilization divided by recurring cost, are tested for deletion. A deletion is accepted only if the full computed score decreases. The previous placement is always an initial incumbent. This provides Proposition 6 for the computed score even though no approximation factor is known.

All tuned variants use 24 evaluations and a four-epoch moving loss window. NoForecast replaces each forecasting slot by the training prior. NoReserve replaces each reserve value by zero. NoSwitch-Penalty replaces every movement weight used in planning by zero, including in cover construction. Disabled values leave duplicate slots. The duplicates are deliberately not converted into extra search over other settings. Real toggle costs remain charged for all variants. A fixed EMA comparison holds

$\epsilon = 0.07$ and $\lambda = 0.95$ and removes only reserve or only the surrogate switching penalty. These nominal settings are specified rather than selected on evaluation outcomes.

The selection loss for a candidate is its complete execution cost divided by $\max\{1, \mathbf{1}^\top n_t\}$. There is no reserve-dependent bonus or subtraction. Counterfactual candidate costs are generated only after the current action was selected. These computations, including unselected candidates, are included in controller runtime.

G.4 Rolling-horizon and full-horizon optimization

The MPC formulation repeats Eq. (3) for three predicted epochs and couples successive binary placements by the toggle cost. It uses a common reserve 0.07 and movement weight 0.95, with recursive last-value, EMA, or ridge predictions. Only the first planned placement is executed, and the next optimization starts from the physical state produced by actual repair. No terminal shutdown is imposed. A solve without an incumbent falls back to the local planner. Every reported medium MPC solve returned an incumbent.

The compact complete-information comparator uses reserve zero, movement weight one, the full realized six-epoch demand, and the same all-closed initial state. Its objective charges direct transitions between consecutive physical placements. Because advance knowledge permits choosing $p_t = q_t$, this is also optimal for the two-stage execution model with nonnegative repair surcharge. All solutions reported in the compact comparison have closed solver gaps to the stated tolerance. The horizon optimum is not estimated from a clairvoyant heuristic.

For medium-instance numerical bounds, the LP relaxation is solved on the current realized demand from the actual previous state. The numerical lower bound is obtained from dual multipliers using Eq. (31). The reported cost gap is state-conditioned, as in Theorem 5, and must not be interpreted as dynamic regret.

G.5 Accelerator calibration and simulator mapping

The accelerator calibration was obtained separately from the CPU placement experiments on one NVIDIA A100-SXM4-40GB. It gives 39.4935 GiB of available device memory, 250.5959 TFLOP/s for a 4096×4096 dense-matrix computation, and transfer rates of 11.5052 GiB/s from host to device and 12.2886 GiB/s in the reverse direction. Transfers use 256 MiB chunks. Memory and transfer quantities are expressed in binary units.

An isolated attention calculation yields 58,655.09 batch elements per second at batch size 8 and sequence length 2,048. It consists of a query-key product, softmax, and attention-value product, not a transformer forward pass or full autoregressive decoding. The available calibration does not include all measurement settings or timing repetitions, so no uncertainty interval is reported for these rates.

The simulator uses the measured dense rate P, memory H, and host-to-device rate B through the dimensionless factor

$$\chi = \operatorname{clip}\left(1.35 + 0.075\sqrt{P} + 0.018\sqrt{H}, 1.35, 3.25\right). \tag{51}$$

A fixed 22% of ordinary bundles, distributed by evenly spaced indices, are accelerator-tagged. Their work capacity is multiplied by χ, occupancy capacity by $1.15+0.08\chi$, recurring cost by $1.35+0.12\chi$, and toggle cost becomes $m_i(1.15 + 0.04\chi) + 20/\max(B, 1)$. Neither the attention rate nor the device-to-host rate enters this mapping.

For code, database, search, media, and github families, accelerator-tagged connection costs are multiplied by $\max(0.34, 0.82/\chi)$, while ordinary-site costs are multiplied by 1.10. Stateful non-compute-heavy types receive factor 0.78 at tagged sites and other types receive 0.92. These coefficients are hand-chosen scenario parameters, not a fitted physical model. The resulting resource units are not literal GPU occupancy or byte-level HBM use. The CPU and calibrated studies also differ in horizon, offered load, and number of bundles. Comparisons across the two studies therefore cannot isolate a causal hardware effect.

G.6 SENSITIVITY DESIGN

Each short stress run uses 12 epochs, 1,000 offered units per epoch, 48 ordinary bundles plus three fallback bundles, and one of the four scenario families. Three seeds are paired across SAGE-R, fixed EMA, and last-value control. All training streams retain nominal settings. The following parameters are varied individually at evaluation time.

Axis	Values	Meaning
Capacity	$0.75, 1, 1.25$	Multiply every resource capacity
Toggle cost	$0.25, 1, 4$	Multiply all toggle fees
Forecast mixing	$0, 0.5, 1$	Mix prediction with a three-type cyclic permutation
Compatibility thinning	$0, 0.25, 0.5$	Delete sampled ordinary compatible edges
Family-shift strength	$0, 1, 3$	Scale the specified family trend
Burstiness	$0.25, 1, 4$	Divide Dirichlet concentration by this factor
Repair surcharge	$0, 20, 100$	Add a fixed charge for each emergency activation

Compatibility thinning preserves the three designated fallback neighborhoods, but their capacities remain finite. Forecast mixing preserves the forecast's total mass, not its compatibility composition. The family-shift knob affects only the specified family trend, not every time-varying term in the generator. The nominal condition is shared across sweeps rather than rerun as an additional independent sample. These experiments locate behavior within the constructed parameter space. They do not show where real workloads fall in that space.

G.7 ACCOUNTING, UNCERTAINTY, AND COMPUTATION

For each epoch, repair-served mass is $\sum_{a,i \mid q_i > p_i} y_{ai}$. The repair attribution includes connection cost of all such assignments, the added bundles' full epoch residency and activation cost, and any surcharge. A retirement decided before demand arrives remains part of planned switching cost. Thus closing and reopening one bundle can incur both a planned retirement charge and a repair opening charge. The total objective includes both, while the repair attribution includes only the latter transition. This convention is uniform across policies.

The optional connection-cost quantile is computed by weighting the actual assigned pair costs c_{ai} by accepted masses y_{ai}. It is a percentile of the modeled connection-cost distribution conditional on acceptance, not of measured request latency. The main comparisons do not use it.

Family-level cost tables take the mean of cost per offered mass over the stated seeds. Paired percentage improvements are computed before averaging, so they need not equal a ratio of two family-average costs. Bootstrap calculations resample seeds separately inside each family. They do not resample families, requests, or epochs as independent experimental units. The small seed count and common generator are explicit limits on inference.

The experiments used Python 3.13.5, NumPy 2.3.5, and SciPy 1.17.0 on a Linux virtual CPU environment exposing five logical cores of an AMD EPYC 9V74 processor. Four worker processes each used one BLAS/OpenMP thread. Timings include concurrent load and should not be interpreted as dedicated-host performance. The A100 calibration supplies coefficients to CPU simulations, and no new accelerator measurement was performed. Further implementation details are in the supplement.

H ADDITIONAL RESULTS

Table 8. Fixed-parameter CPU controls. All rows use the same local-search and execution routines. Bold marks the lowest mean among these displayed controls, not a significance claim.

Policy	API	Tool	MCP	App
EMA-Fixed	1.669	2.073	**1.751**	1.938
Last-Fixed	1.688	2.114	1.772	1.925
Prior-Fixed	1.722	2.172	1.832	1.975
Ridge-Fixed	1.715	2.161	1.820	1.938
EMA-NoReserve	1.672	2.131	1.797	1.940
EMA-NoSwitchPenalty	**1.669**	**2.050**	1.787	**1.910**

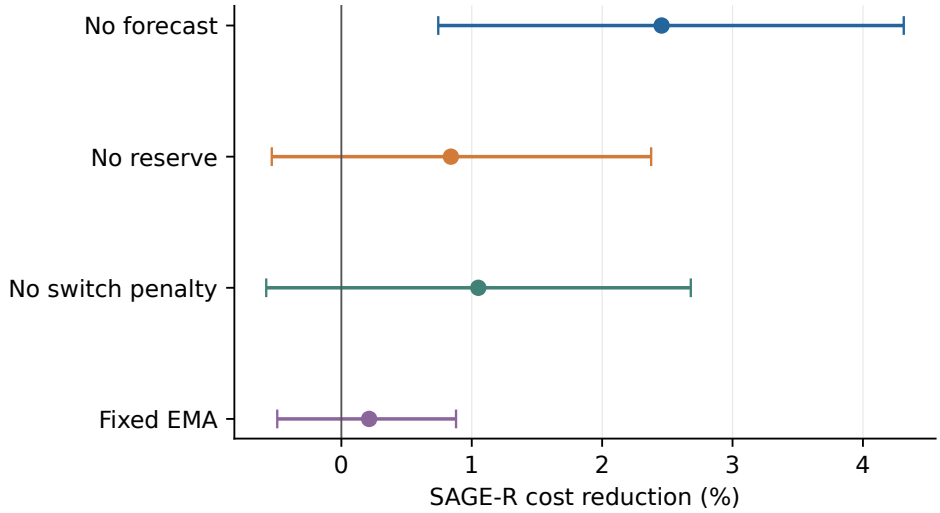

Figure 7. Matched-budget component comparisons. Positive values favor SAGE-R. Bars give within-family bootstrap intervals. The no-forecast control differs from fixed EMA and should not be interpreted as the strongest forecasting baseline.

Table 9. Compact full-horizon comparison. Entries average two ratios to the reported horizon solver reference per family. These are new instances and new controllers. Lower is better.

Policy	API	Tool	MCP	App
SAGE-R	1.299	1.085	1.094	1.053
EMA-Fixed	1.299	1.085	1.094	1.053
MPC-EMA	**1.000**	**1.002**	1.012	1.004
MPC-Ridge	**1.000**	1.006	**1.004**	**1.004**

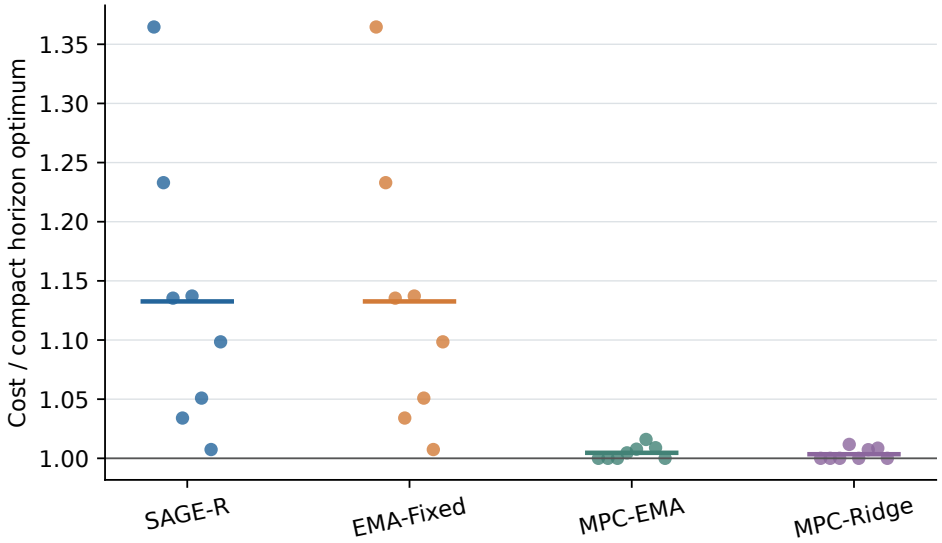

Figure 8. All eight compact instances. Horizontal segments mark policy means. Identical SAGE-R and EMA values in this compact study reflect identical selected decisions on these small instances, not an independently counted baseline. The policies remain distinct on the main suites.

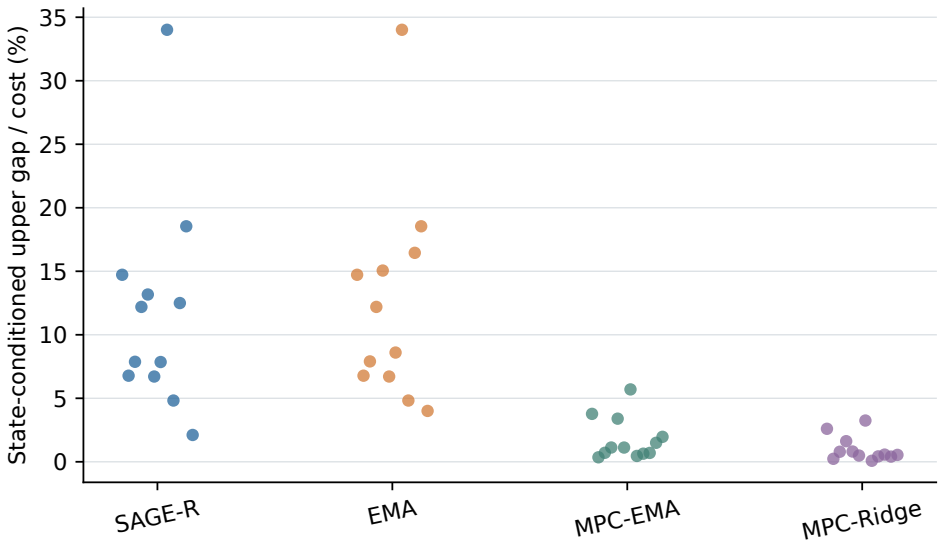

Figure 9. Medium-study one-step numerical bounds. Each point is a run's summed upper bound on state-conditioned one-step losses divided by its total cost. The comparator starts from that policy's actual previous state at each epoch. This quantity is not a full-horizon optimality gap.

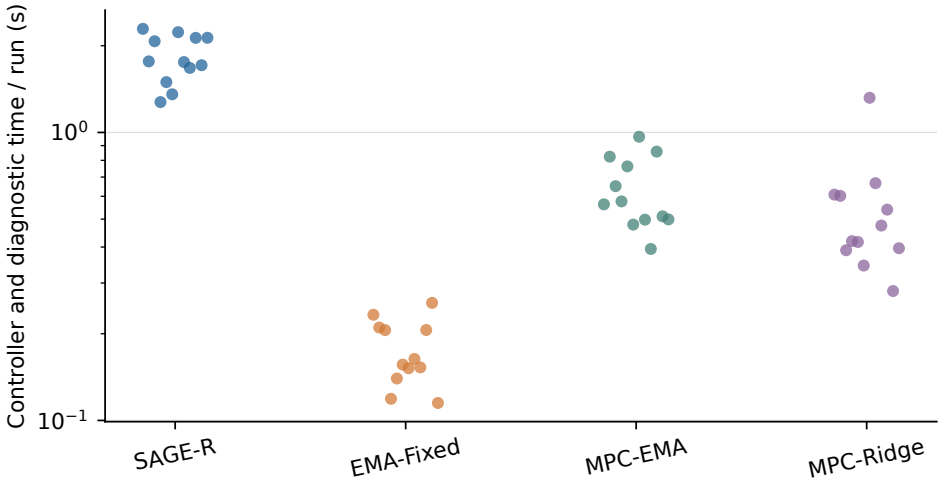

Figure 10. Wall time for a complete medium run, including historical scoring and posterior diagnostics. Local portfolios and MPC perform different computations. No bar represents serving latency or inference speedup.

H.1 CALIBRATED SIMULATION AND PAID REPAIR

The family averages and all seed effects for the calibrated study appear in Table 6 and Figure 6. The following tables separate paid repair from planned placement under the same accounting rules. CPU accounting is reported in Table 7.

Table 10. Calibrated-simulation repair accounting under the same definitions. Zero compatibility violations do not imply zero rejection or optimal placement.

Policy	Repair mass	Repair cost	Rejection	Closures
SAGE-R	0.407%	0.201%	1.239%	3484
Last-Fixed	0.347%	0.160%	1.268%	7114
EMA-Fixed	0.293%	0.143%	1.251%	4079
Ridge-Fixed	0.342%	0.171%	1.245%	819

H.2 INDEPENDENT PARAMETER SWEEPS

All following panels compare SAGE-R with fixed EMA on the same short scenarios and seeds. Lines are family means and small points are the individual seed effects. Positive values favor SAGE-R. The comparison is conditional on these generated workloads.

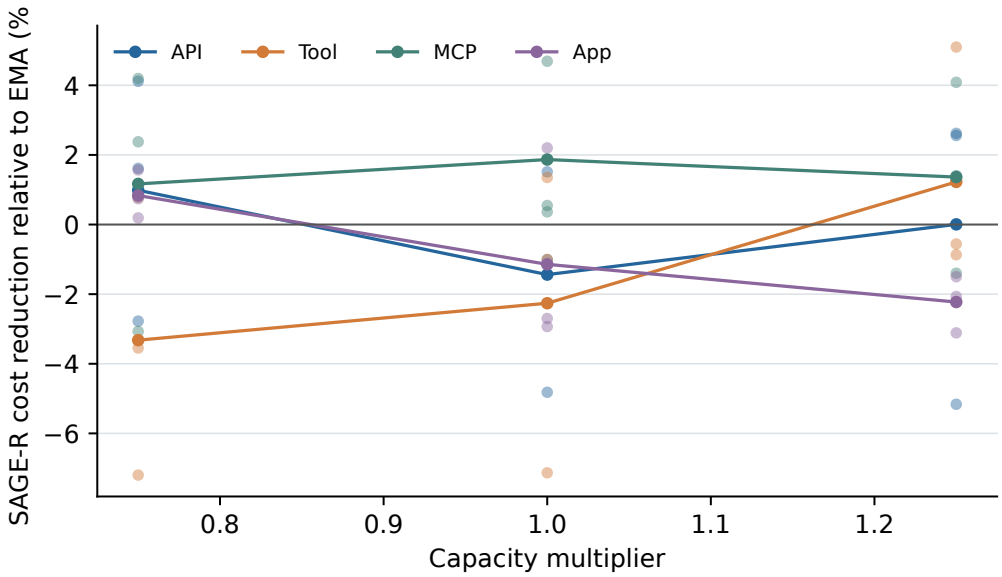

Figure 11. Capacity scaling applied to every resource and bundle.

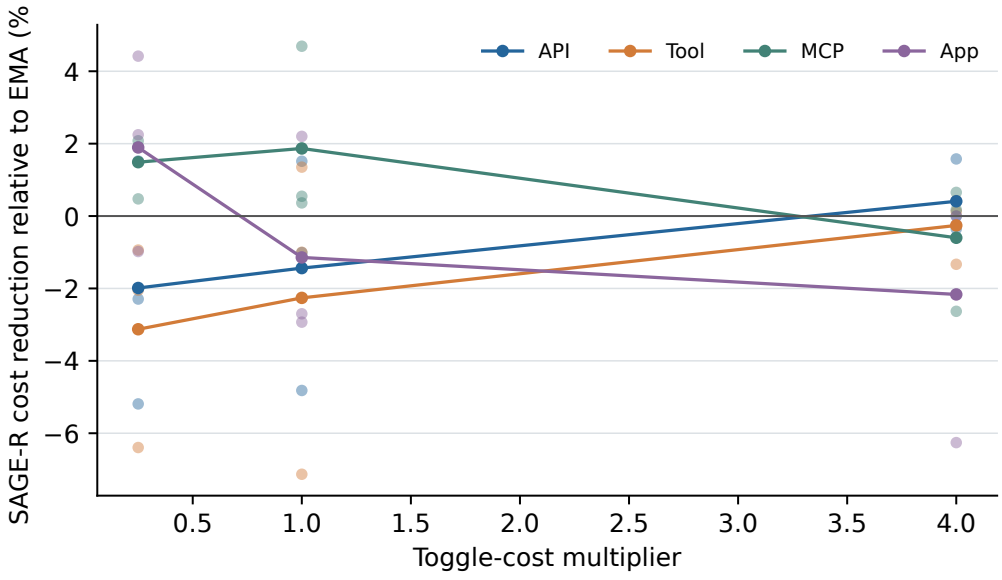

Figure 12. Real toggle costs and their corresponding planner coefficients are scaled together.

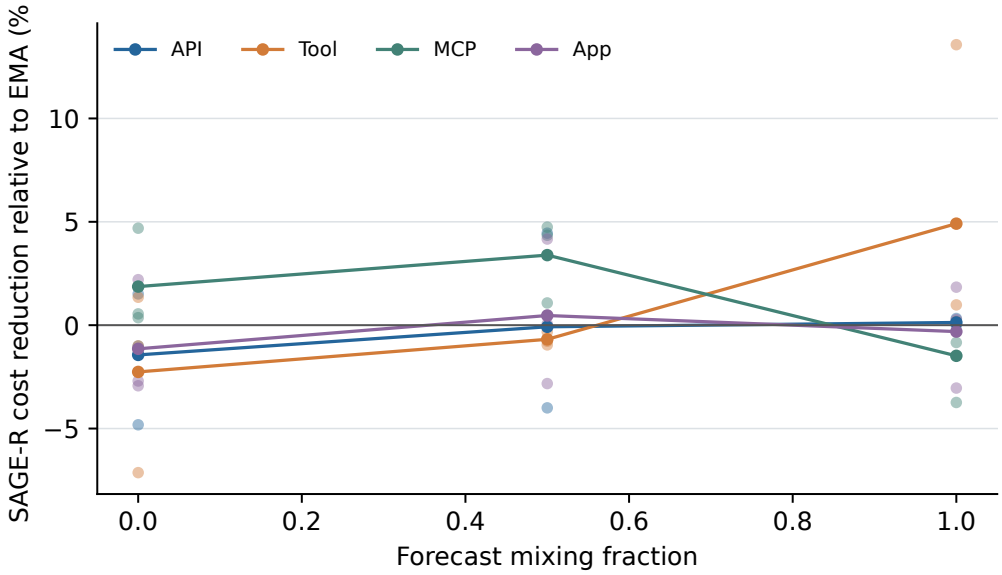

Figure 13. Forecast mixing across type labels. Both compared controllers receive the same perturbation operator. Larger mixing is not asserted to induce a monotone increase in decision loss.

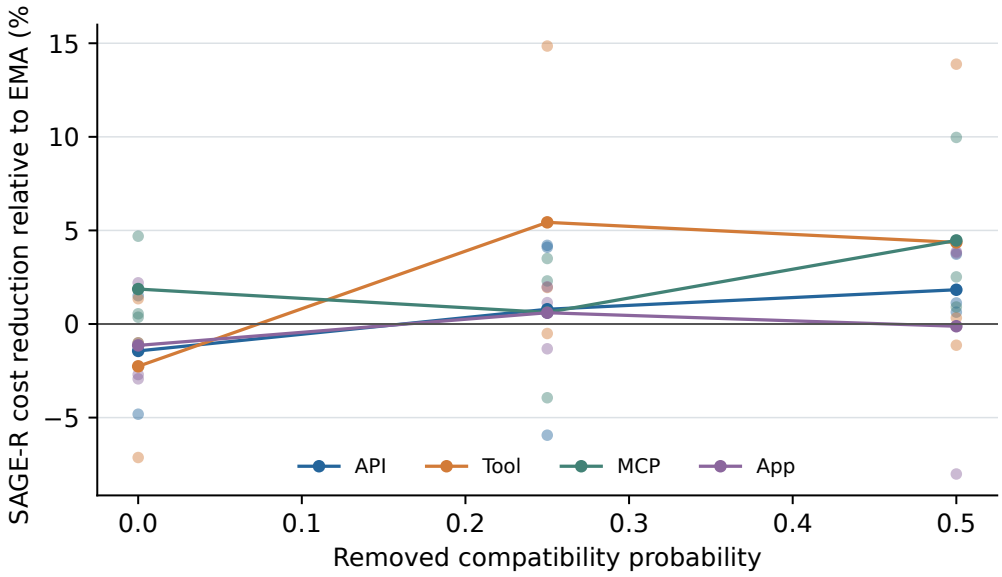

Figure 14. Thinning ordinary compatibility neighborhoods while retaining finite fallback neighborhoods.

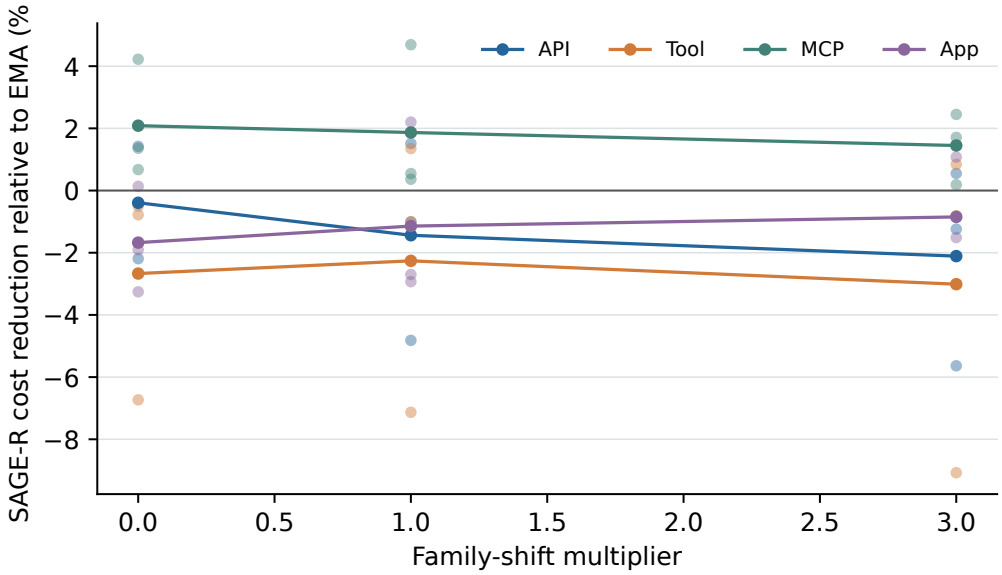

Figure 15. Magnitude of the specified family-demand trend.

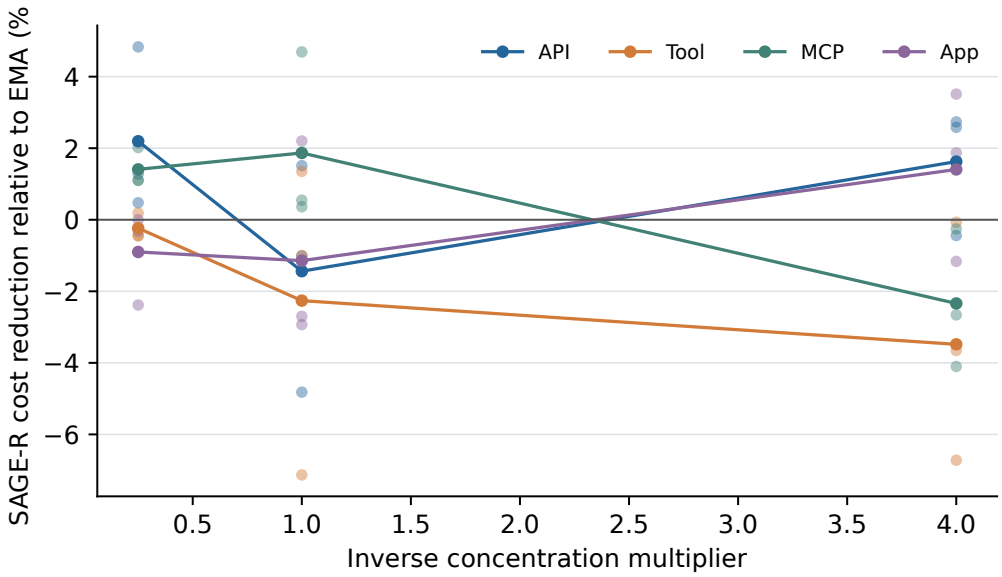

Figure 16. Count dispersion changed through Dirichlet concentration while the deterministic trend remains fixed.

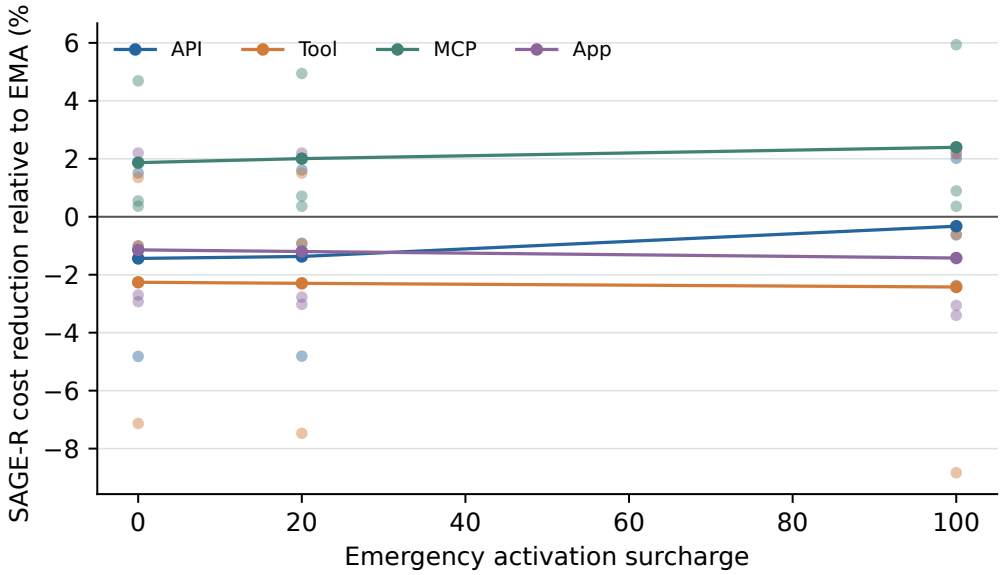

Figure 17. Additional charges for post-observation activation. Preplanned openings do not incur this surcharge.