COMPRESSING VALUE PREDICTIONS FOR LEARNING-AUGMENTED METRICAL TASK SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning-augmented algorithms for metrical task systems (MTS) can exploit predictions of canonical dual values, but existing formulations typically require a prediction for every state. We study whether these predictions can be compressed to a small set of representative states while retaining their algorithmic value. We introduce *landmark-compressed value predictions*, in which the predictor reports predicted dual values only at m landmarks and the remaining values are reconstructed by a Lipschitz extension. Our algorithm achieves additive excess cost $O(T\, r(L) + \sum_t \delta_t)$, where $r(L)$ is the covering radius of the landmarks and δ_t measures prediction error up to additive shifts; local and value-dependent bounds refine this guarantee. For sparse landmark sets on unit-spaced finite lines, we prove a matching $\Omega(T r_m)$ lower bound for every randomized algorithm using fixed landmarks, even with advance access to their entire exact absolute-value table. The prediction interface also matters: on two states with one landmark, exact absolute values permit horizon-independent excess, whereas exact relative values force worst-case expected excess linear in T. We give PAC guarantees for learning compressed prediction tables, with efficient empirical-risk minimization for fixed landmarks. Our results connect metric coverage, prediction interfaces, and learning guarantees for compressed predictions in online MTS.

1 INTRODUCTION

Online optimization studies problems in which inputs arrive sequentially and each decision must be made before future inputs are known. Metrical task systems (MTS) capture a basic tradeoff in such decisions (Borodin et al., 1992): at each round, an algorithm chooses a state and pays both its service cost and the cost of moving there from the previous state. Tasks can favor different states over time. Moving to a cheaper state may save service cost now, yet require costly moves later; staying avoids movement but may incur higher service costs.

Classical competitive analysis compares an online algorithm with the offline optimum, which knows the entire input sequence in advance (Borodin & El-Yaniv, 1998). An algorithm is α-competitive if, on every sequence, its cost is at most α times the optimum, up to an input-independent additive constant; randomized algorithms are evaluated in expectation. This provides guarantees against arbitrary future inputs. In applications with recurring task patterns, however, historical data can help anticipate future demand. Learning-augmented algorithms use such predictions to improve decisions, while analyzing how their performance depends on prediction quality (Lykouris & Vassilvitskii, 2018; Purohit et al., 2018). For MTS, this approach has been developed using predictions of the states to visit (Antoniadis et al., 2023b).

For MTS, canonical dual values give each state's optimal cost of serving future tasks after the current round. Adding them to immediate movement and service costs identifies an optimal next action when the values are exact. Coester et al. (2026) establish performance and learning guarantees for full prediction tables using a value-greedy policy. Building on this policy, we study the information lost when canonical values are available only at a fixed set of representative states, called *landmarks*. We show that the resulting covering-radius dependence is order-optimal in the sparse finite-line regime, even with advance access to the entire exact absolute-value table on the landmarks. We also show that absolute and relative value interfaces can yield qualitatively different guarantees.

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.

A full predictor on an n-state metric reports n predicted dual values per round; storing and fitting an explicit table requires a parameter for every state and prediction time. This scaling motivates reducing the representation size. Generating predictions can also incur computational costs, motivating algorithms that use predictions sparingly. In caching, Im et al. (2022) reduce the number of next-arrival-time predictions, while Sadek & Eliáš (2024) study fewer calls to an action predictor for caching and MTS. We study the complementary question of reducing the number of predicted values per round while keeping all states available to the online algorithm. The metric relates values at nearby states, suggesting redundancy. The challenge is to control the cumulative decision cost caused by reconstruction errors, and to determine which losses are unavoidable for any algorithm with the same limited information. These observations lead to our central question:

What is the tradeoff between prediction dimension and online performance in metrical task systems?

To study this tradeoff, we retain predicted dual values only at a fixed set L of m landmarks, selected before observing the test tasks or their predictions. The algorithm observes all current service costs and can act at any state. Future values are 1-Lipschitz, allowing unreported values to be estimated by a metric extension (McShane, 1934) and used in a value-greedy decision rule. With exact values at every state, this rule achieves the offline optimum. We therefore measure the extra cost introduced by compression and prediction errors through the additive excess $\mathrm{ALG} - \mathrm{OPT}$. The greedy rule depends only on relative values within each round.

Our contributions.

- **Compression guarantees.** For horizon T, our value-greedy policy satisfies
$$\mathrm{ALG} - \mathrm{OPT} \le 2(T-1)r(L) + 2\sum_{t=1}^{T-1} \delta_t,$$
where $r(L)$ is the maximum distance to the nearest landmark and δ_t is the maximum absolute landmark prediction error, minimized over common additive shifts. Metric coverage thus controls the geometric compression error term; local successor-radius and value-dependent bounds refine this guarantee.
- **Matching lower bound for fixed-landmark access.** On unit-spaced finite lines in the sparse regime of Theorem 3, every randomized algorithm using at most m fixed landmarks has worst-case expected excess $\Omega(Tr_m(M))$, even given the entire exact absolute-value table on those landmarks in advance. Here $r_m(M)$ is the optimal m-covering radius. This matches our upper guarantee and limits every policy with this information access, regardless of its reconstruction rule.
- **Prediction-interface separation.** On two states with one landmark, exact absolute values permit excess at most $2r(L)$, whereas exact relative values force $\Omega(Tr(L))$ worst-case expected excess for every randomized policy (Theorem 4). More generally, exact absolute values at all but one state permit excess at most $2r(L)$. Thus the absolute cost scale can help other policies, although common shifts do not affect our greedy decisions.
- **Learnable representations.** Using objectives derived from our performance bounds, we give PAC guarantees and a polynomial-size linear program for fitting tables on fixed landmarks, plus generalization guarantees for joint landmark and table selection. The learned tables reuse the same time-indexed vectors across test episodes. A Citi Bike-derived K-server benchmark evaluates prediction-budget tradeoffs for mean-table and causal neural predictors.

2 Problem Setup

Metrical task systems. Let (M, d) be a finite metric space with $n = |M| \ge 2$ and diameter D. The initial state $s_0 \in M$ and horizon $T \ge 2$ are known. An episode $X = (c_1, \ldots, c_T)$ consists of finite service costs $c_t : M \to \mathbb{R}_{\ge 0}$. At round t, the algorithm observes all of c_t, chooses $s_t \in M$, and pays $d(s_{t-1}, s_t) + c_t(s_t)$. Write ALG for its total cost and OPT for the minimum cost of any path starting at s_0 with knowledge of X. We study additive excess $\mathrm{ALG} - \mathrm{OPT}$.

Canonical dual values. Let $w_t(a)$ denote the minimum cost of serving the remaining tasks $c_{t+1}, \ldots, c_T$, starting from state a *after* round t. No tasks remain after round T, so $w_T \equiv 0$.

The other values are determined backward by the Bellman recursion:

$$w_T \equiv 0, \qquad w_{t-1}(a) = \min_{b \in M}\{d(a,b) + c_t(b) + w_t(b)\}. \tag{1}$$

In particular, $w_0(s_0) = \mathrm{OPT}$.

These future costs have a dual interpretation. An offline MTS episode is a shortest-path problem on a layered graph, where the edge $(t-1, a) \to (t, b)$ has cost $d(a,b) + c_t(b)$. The corresponding unit-flow LP has dual potentials $u_t(a)$. With $u_T \equiv 0$, the dual maximizes $u_0(s_0)$ subject to $u_{t-1}(a) \leq d(a,b) + c_t(b) + u_t(b)$ for all t, a, b (Coester et al., 2026). The Bellman values satisfy these constraints and attain OPT, so they form an optimal dual solution. The recursion specifies these values uniquely; we use this canonical solution as our prediction target.

We write $(B_c f)(a) = \min_b\{d(a,b) + c(b) + f(b)\}$ for the Bellman operator. The triangle inequality implies that each w_t is 1-Lipschitz, i.e., $|w_t(a) - w_t(b)| \leq d(a,b)$ (Lemma 6, Appendix A.1). Thus a landmark's value constrains values at nearby states.

Landmarks and predictions. Fix a nonempty landmark set $L \subseteq M$, with $|L| = m$. The covering radius $r(L)$ is the largest distance from a state to its nearest landmark, and $r_m(M)$ is the smallest radius achievable with at most m landmarks:

$$d(x, L) = \min_{\ell \in L} d(x, \ell), \qquad r(L) = \max_{x \in M} d(x, L), \qquad r_m(M) = \min_{\substack{S \subseteq M \\ 1 \leq |S| \leq m}} r(S). \tag{2}$$

Before choosing s_t at each round $1 \leq t < T$, the algorithm receives predictions $v_t \in \mathbb{R}^L$ of $w_t|_L$, with one predicted value per landmark; $w_T \equiv 0$ is known exactly. Our value-greedy policy in Section 3 depends only on relative values within each round. With $\|z\|_{\mathrm{sp}} = \max z - \min z$, we therefore measure prediction error modulo a common additive shift:

$$\delta_t = \inf_{a \in \mathbb{R}} \max_{\ell \in L} |v_t(\ell) - w_t(\ell) - a| = \tfrac{1}{2}\|v_t - w_t|_L\|_{\mathrm{sp}}. \tag{3}$$

The span identity is proved in Appendix A.2.

Prediction access model. The landmark set L is selected before any service costs or predictions from the episode are observed and remains fixed throughout the episode. We distinguish two interfaces for exact values: the *Raw* interface reveals $w_t|_L$, while the *Relative* interface reveals $w_t(\ell) - w_t(\ell_0)$ for a fixed anchor $\ell_0 \in L$. These interfaces can yield different guarantees for general policies, as shown in Theorem 4.

3 Compressing Value Predictions

We quantify the cost of replacing full-state value predictions by landmark predictions. The resulting excess is controlled by metric coverage and landmark prediction error.

Reconstructing from landmarks. The canonical dual values w_t (optimal remaining costs) are 1-Lipschitz with respect to d. Thus each exact landmark value gives an upper bound $w_t(x) \leq w_t(\ell) + d(x, \ell)$, and taking the minimum over landmarks gives the tightest of these bounds. For landmark predictions v_t at round $t < T$, define the reconstruction operator E_L and the reconstructed values $\widehat{w}_t$ by the metric envelope

$$\widehat{w}_t(x) \coloneqq (E_L v_t)(x) = \min_{\ell \in L}\big\{v_t(\ell) + d(x, \ell)\big\}. \tag{4}$$

With exact landmark predictions, the reconstructed values upper-bound w_t; arbitrary landmark predictions need not preserve this property. The value-greedy policy selects

$$s_t \in \arg\min_{x \in M} \left\{d(s_{t-1}, x) + c_t(x) + \widehat{w}_t(x)\right\}. \tag{5}$$

At the terminal round we set $\widehat{w}_T = w_T \equiv 0$. A common additive shift of v_t produces the same shift in $E_L v_t$, so the decisions made by (5) depend only on relative landmark predictions.

The reconstruction error is controlled by the distance to the nearest landmark and the accuracy of the landmark predictions.

Lemma 1 (Landmark reconstruction)**.** *For exact landmark values,*

$$0 \le E_L(w_t|_L)(x) - w_t(x) \le 2d(x, L) \le 2r(L). \tag{6}$$

For arbitrary landmark predictions with per-round landmark prediction error δ_t defined in (3)*,*

$$\left\|\widehat{w}_t - w_t\right\|_{\mathrm{sp}} \le 2r(L) + 2\delta_t. \tag{7}$$

The exact bound (6) follows directly from Lipschitzness: every landmark gives an upper bound on $w_t(x)$, while the nearest landmark is at most $2d(x, L)$ above it. For arbitrary landmark predictions, let a_t and b_t be the minimum and maximum of $v_t(\ell) - w_t(\ell)$ over $\ell \in L$. Taking minima in (4) preserves these bounds, so for every x,

$$a_t \le E_L v_t(x) - E_L(w_t|_L)(x) \le b_t.$$

Hence replacing exact landmark values by predictions adds at most $b_t - a_t = 2\delta_t$ to the reconstruction error span, proving (7) by the span triangle inequality. Appendix B gives the full reconstruction argument.

Our main upper bound follows by relating the reconstruction error span to the Bellman gap at each round.

Theorem 2 (Compressed value prediction)**.** *For any fixed landmark set L and arbitrary landmark predictions $\{v_t\}_{t<T}$, the value-greedy policy in* (5) *satisfies*

$$\mathrm{ALG} \le \mathrm{OPT} + 2(T-1)r(L) + 2\sum_{t=1}^{T-1} \delta_t. \tag{8}$$

In particular, with exact landmark predictions,

$$\mathrm{ALG} - \mathrm{OPT} \le 2(T-1)r(L).$$

Proof sketch. Let $e_t = \widehat{w}_t - w_t$ denote the reconstruction error, and let $x_t^\star$ be a Bellman-optimal successor from the algorithm's current state s_{t-1}. Define the Bellman gap incurred by the algorithm at time t as

$$\Delta_t = d(s_{t-1}, s_t) + c_t(s_t) + w_t(s_t) - w_{t-1}(s_{t-1}).$$

These gaps are nonnegative and telescope to $\mathrm{ALG} - \mathrm{OPT} = \sum_t \Delta_t$. Since s_t minimizes the predicted objective in (5), whereas $x_t^\star$ minimizes the true Bellman objective,

$$\begin{aligned} \Delta_t &\le e_t(x_t^\star) - e_t(s_t) \\ &\le \|e_t\|_{\mathrm{sp}}. \end{aligned} \tag{9}$$

The reconstructed values at round T are exact, so $\Delta_T = 0$. Summing (9) over $t < T$ and applying Lemma 1 proves (8). The complete argument is given in Appendix B.

Interpretation. The bound separates the geometric compression error term $2(T-1)r(L)$ from the prediction error term $2\sum_{t<T}\delta_t$. With exact landmark predictions and landmarks attaining the optimal covering radius $r_m(M)$, the excess cost is at most $2(T-1)r_m(M)$. Thus metric covering complexity controls the prediction budget sufficient for a target excess cost.

Reconstruction distortion. The geometric compression error bound $2r(L)$ depends only on landmark coverage. To account for the values of an episode X, define its reconstruction distortion as

$$\kappa_t(L; X) \coloneqq \|E_L(w_t^X|_L) - w_t^X\|_{\mathrm{sp}} \le 2r(L).$$

The same Bellman-gap argument gives

$$\mathrm{ALG} - \mathrm{OPT} \le \sum_{t<T} \left[\kappa_t(L; X) + 2\delta_t\right],$$

as proved in Appendix B.3. The distortion can vanish even for a large covering radius: if $w_t^X(x) = d(x, a)$ and $L = \{a\}$, then $\kappa_t(L; X) = 0$. Since κ_t depends only on L and the episode's exact values, it can be computed from offline training labels independently of the predictions and online trajectory. Section 5 combines this distortion with the prediction error in its joint landmark and table selection objective.

The next section asks whether the worst-case dependence on $r_m(M)$ is unavoidable.

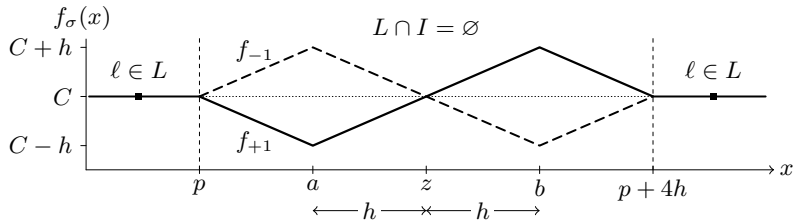

Figure 1: Hidden value profiles on an unhit interval: identical landmark values, opposite favorable candidates. The common suffix K_j in $w_{4j-2} = f_\sigma + K_j$ is omitted. Lines interpolate discrete-state values; the text establishes whole-table indistinguishability.

4 Lower Bounds and Prediction Interfaces

We now show that the dependence on the covering radius in our upper bound is unavoidable when access to future values is restricted to fixed landmarks. The lower bound already holds in a stronger prediction model than required by our algorithm: after choosing a fixed landmark set L, the online algorithm may receive the exact absolute values $\{w_t(\ell)\}_{\ell \in L}$, and may even inspect the entire table of landmark values before the episode starts. We prove the result on the finite line $M = \{0, \dots, n-1\}$ with $d(x, y) = |x - y|$, using the optimal covering radius $r_m(M)$ defined in (2).

Theorem 3 (Lower bound for fixed landmarks)**.** *Let $M = \{0, \dots, n-1\}$ be the unit-spaced line. If $n - 1 \geq 16m$ and $T \geq 4$, then for every randomized online algorithm using at most m fixed landmarks, there exists an oblivious task sequence such that*

$$\mathbb{E}[\mathrm{ALG}] - \mathrm{OPT} \geq \frac{1}{32} \lfloor \frac{T}{4} \rfloor \, r_m(M).$$

The bound holds even when the entire table of exact absolute landmark values is revealed after L is selected. Together with the $O(Tr_m(M))$ upper bound, this gives minimax excess $\Theta(Tr_m(M))$ for access to exact values at fixed landmarks in this sparse finite-line regime.

Proof idea. Each four-round gadget hides which of two states is favorable inside one of $2m$ disjoint intervals. At least half of these intervals contain no landmark. On such an interval, the algorithm cannot identify the favorable state, even from the entire landmark-value table, and incurs expected excess proportional to the interval's length. Repeating the gadget makes these losses accumulate. We sample all intervals and hidden signs independently of the algorithm's randomness, including its choice of L. Appendix C gives the complete construction.

Let $D = n - 1$ and $h = \lfloor D/(8m) \rfloor$. Place $2m$ disjoint open intervals $I_i = (4hi, 4h(i+1))$, $i = 0, \dots, 2m-1$, on the line. Each gadget independently samples one interval uniformly, so, conditional on any realized L, its interval is unhit with probability at least $1/2$. Within a selected interval $I = (p, p + 4h)$, set $a = p + h$, $z = p + 2h$, and $b = p + 3h$. An independent fair sign $\sigma \in \{-1, +1\}$ determines which candidate has the smaller future value. We construct a nonnegative 1-Lipschitz function f_σ with a common baseline C such that

$$f_\sigma(a) = C - \sigma h, \qquad f_\sigma(b) = C + \sigma h, \qquad f_\sigma(x) = C \quad (x \notin I).$$

The gadget has four stages: enter z, decide between the candidates, reveal σ, and reset to the common origin $o = s_0$. Finite service penalties make the enter and reset actions uniquely Bellman-optimal. The decision task c^{dec} costs zero at a, b and outside I, and penalizes all other interior states. The reveal and reset tasks realize f_σ as the values after the decision task, up to the optimal cost of subsequent gadgets. Figure 1 shows how the favorable candidate changes while landmark values remain identical.

To hide σ from the *entire* table, we must also prevent it from changing earlier absolute values through the gadget's optimal cost. Writing $g_\sigma = B_{c^{\mathrm{dec}}} f_\sigma$, the construction ensures

$$g_\sigma(z) = C, \qquad g_\sigma(x) = C \quad (x \notin I).$$

Thus the optimal cost of a complete gadget starting at o is $d(o, z) + C$, independently of its sign. The optimal cost K_j of all gadgets after gadget j is consequently independent of every hidden sign.

Within gadget j, the only value functions that can depend on its sign are $g_\sigma + K_j$ after the enter task and $f_\sigma + K_j$ after the decision task. Both equal $C + K_j$ at every landmark when $L \cap I = \varnothing$, and no sign information propagates to earlier values. Since the observed tasks through the decision stage are also independent of σ, it remains a fair hidden bit at that stage.

If the algorithm skips the enter action to z, its Bellman gap is already at least h. Otherwise, both candidates are distance h from z: the favorable one has gap zero and the other has gap $2h$, giving expected gap h for either choice. Other actions cannot avoid this loss: an exterior state has gap at least $2h$, and the penalty at any other interior state gives gap at least h. Hence every randomized decision rule satisfies

$$\mathbb{E}[\Delta_{\text{in}} + \Delta_{\text{dec}} \mid L \cap I = \varnothing] \geq h.$$

All remaining Bellman gaps are nonnegative. Since each interval is unhit with probability at least $1/2$, each gadget contributes expected excess at least $h/2$. Concatenating $N = \lfloor T/4 \rfloor$ gadgets and appending zero tasks to reach horizon T gives

$$\mathbb{E}[\text{ALG} - \text{OPT}] \geq \frac{Nh}{2} \geq \frac{N}{32}\, r_m(M),$$

where $h \geq D/(16m) \geq r_m(M)/16$ in the stated sparse regime. Here the expectation includes both input and algorithm randomness. Averaging over inputs fixes an oblivious task sequence with the same lower bound in expectation over the algorithm's randomness, proving Theorem 3. □

The theorem concerns fixed landmarks in the sparse finite-line regime; adaptive queries and arbitrary encodings of future information are outside its scope.

Absolute versus relative values. Outside the sparse regime, the Raw interface can support horizon-independent excess. The following separation uses the same metric and landmark budget for both interfaces.

Theorem 4 (Raw versus Relative interfaces)**.** *On a two-state metric of distance $a > 0$ with one fixed landmark, round-by-round access to exact Raw values permits a deterministic policy with excess at most $2a$ on every episode, without being supplied* OPT*. For every randomized policy using the exact Relative interface, there is an oblivious episode with*

$$\mathbb{E}[\text{ALG}] - \text{OPT} \geq \frac{a}{2}\left\lfloor \frac{T}{4} \right\rfloor. \tag{10}$$

The Raw policy initializes a budget $U \in [\text{OPT}, \text{OPT} + 2a]$ using reconstructed values at the first round. The unspent budget always covers the true remaining cost: if the landmark's movement, service, and exact remaining cost fit, the policy selects the landmark; otherwise, Bellman optimality makes the sole unreported state feasible. With one landmark, the Relative value is always zero, so the two-state construction can hide a fresh favorable state in each four-round gadget. Appendix D gives both proofs and extends the Raw guarantee to $2r(L)$ whenever only one state value is unreported.

Absolute values therefore carry information that general policies can use across rounds, even though common shifts do not affect value-greedy decisions. Theorem 3 shows that the sparse finite-line obstruction persists even when this information is supplied for every round in advance.

5 Learning Landmarks and Prediction Tables

We learn from N_{train} independent episodes drawn from $\mathcal{D}$; tasks within an episode may be dependent. Offline Bellman recursion supplies exact labels. Set $\tau = T - 1$. A *fixed prediction table* reuses the same v_t at time t in every test episode.

Learning a fixed table. Fix L and an anchor $\ell_0 \in L$. The cost bound suggests the span risk

$$q_t^X(\ell) = w_t^X(\ell) - w_t^X(\ell_0), \qquad R_L(v) = \mathbb{E}_{X\sim\mathcal{D}}\left[\frac{1}{\tau}\sum_{t=1}^{\tau} \|v_t - q_t^X\|_{\text{sp}}\right]. \tag{11}$$

Each summand equals $2\delta_t$. Without losing decisions or increasing span loss, we restrict to the compact class $\mathcal{V}_L$ of 1-Lipschitz tables anchored by $v_t(\ell_0) = 0$ (Appendix E.1). Empirical risk minimization (ERM) replaces the expectation by the training average and is a polynomial-size linear program for fixed L (Appendix E.3).

Theorem 5 (Learning a fixed landmark table)**.** *For $\varepsilon > 0$ and $\beta \in (0, 1)$, if*

$$N_{\text{train}} \geq \frac{32D^2}{\varepsilon^2}\left[(m-1)\log\left(1+\frac{32D}{\varepsilon}\right)+\log\frac{2\tau}{\beta}\right], \tag{12}$$

then, with probability at least $1-\beta$ over training episodes, any ERM $\widehat{v} \in \mathcal{V}_L$ satisfies $R_L(\widehat{v}) \leq \inf_{v\in\mathcal{V}_L} R_L(v) + \varepsilon$ and

$$\mathbb{E}_X[\text{ALG}_{L,\widehat{v}} - \text{OPT}] \leq \tau\left[2r(L) + \inf_{v\in\mathcal{V}_L} R_L(v) + \varepsilon\right]. \tag{13}$$

The expectation is over a fresh test episode. For fixed landmarks, the dimension term in (12) is $m-1$, rather than $n-1$ for $L = M$. Each training episode supplies labels at every prediction time; uniform convergence is established per time and then averaged, with a union bound contributing only $\log\tau$. The accuracy ε concerns time-averaged span loss; total statistical excess ε_{tot} requires $\varepsilon = \varepsilon_{\text{tot}}/\tau$ (proof in Appendix E.2).

Selecting landmarks. Unlike fitting v for fixed L, selecting L changes both reconstruction distortion and prediction error. We therefore minimize the empirical version of

$$J(L, v) = \mathbb{E}_X\left[\frac{1}{\tau}\sum_{t=1}^{\tau}\left(\kappa_t(L; X) + \|v_t - q_t^X\|_{\text{sp}}\right)\right], \tag{14}$$

which bounds expected excess by $\tau J(L, v)$. The same LP applies for each fixed L, but joint search is combinatorial. Appendix E.4 gives the joint fixed-table guarantee, including empirical optimization error. Alternatively, independent validation can select among frozen training-fitted candidates by actual excess, with a guarantee relative to the best candidate (Appendix E.5).

6 Experiments

Setup. We derive a finite-state K-server benchmark from Citi Bike trip starts (Citi Bike, 2026). Each day is a separate episode with 96 15-minute slots or 1,440 1-minute slots. We map trip starts to ten ordered latitude bins and replace each nonempty slot by a single request to the bin with the most starts; empty slots require no service. A configuration places K servers in distinct bins, and serving a request requires a server at its bin. The cost of changing configurations is the minimum total distance the servers must move, measured in bin-index units. We evaluate $K \in \{3, 4, 5, 6\}$, giving $n = \binom{10}{K}$ configurations.

At each slot, we predict the canonical dual value for each configuration, namely its optimal remaining cost after the current slot. We retain these predictions at $m = |L|$ landmark configurations and reconstruct the remaining values using the McShane extension before the value-greedy decision. Thus the horizontal axis is $100m/n$: for $K = 4$, 20% means retaining 42 of 210 predicted values per slot. Greedy distortion-based selection (*Distortion*) reduces the reconstruction-error span (maximum minus minimum) on exact training-day dual values computed by dynamic programming. It targets the distortion term in (14); predictors are fitted separately. *Geometric* chooses configurations farthest from their nearest selected landmark; *Random* uses nested random subsets. Landmark sets are selected offline and fixed across slots and evaluation days. The three methods share the same predictor and decision rule. *Full* uses all n predicted values and retains the predictor's errors. Our default mean predictor averages exact training-day dual values by clock slot; the GRU (Cho et al., 2014) uses observed requests and calendar features. All predictors yield spatially 1-Lipschitz outputs, enforced structurally for neural models. Work Function Algorithm (WFA) (Koutsoupias & Papadimitriou, 1995) and Double Coverage (DC) (Chrobak et al., 1991) are prediction-free online baselines.

After chronological validation, we refit predictors on 2023–2024 and evaluate on all 365 days of 2025. We report mean daily ALG/OPT, averaging over seeds within each day before averaging across days; OPT is computed by exact dynamic programming. Protocol details are in Appendices F.5 and F.6.

Compression performance. Retaining approximately 20% of predicted dual values, *Distortion* achieves lower mean daily ALG/OPT than Random, Geometric, WFA, and DC in all eight settings

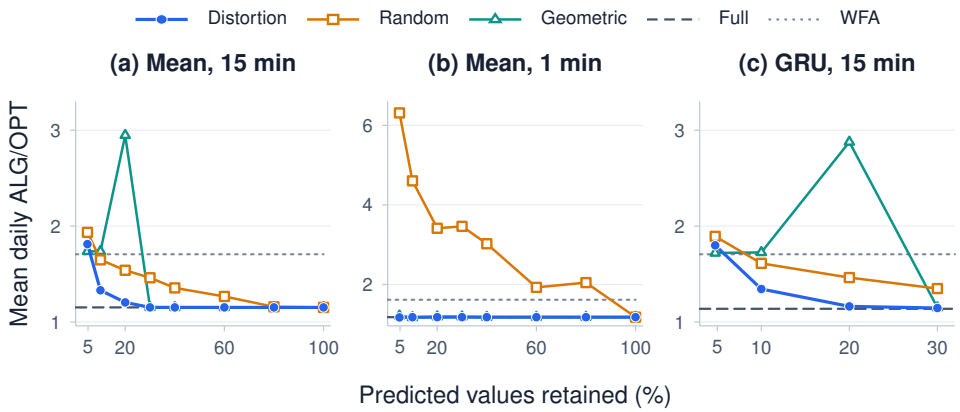

Figure 2: Budget–cost curves for $K = 4$: the mean predictor at two resolutions and a causal GRU. Lines connect tested budgets; axis ranges differ. The GRU's Full endpoint is omitted. Complete results are in Appendix G.

(Table 1). At one minute, its mean ratio differs from Full by less than 0.002 for every tested server count, preserving nearly the same observed performance with one fifth of the values. At 15 minutes, the effect depends more strongly on the setting and budget. For $K = 4$, increasing the prediction fraction from 20% to 30% reduces the ratio from 1.2043 to 1.1527, close to Full's 1.1524 (Figure 2(a)). Compression can also improve observed performance: for $K = 5$, the 20% ratio is 1.2737 compared with Full's 1.3810. Since Full uses imperfect predictions, reconstruction can change its decisions beneficially; retaining more predicted values therefore need not monotonically reduce cost (Appendices G.1 and G.3).

Table 1: Mean daily ALG/OPT on 365 dates with the mean predictor and approximately 20% of predicted values retained. Bold marks row minima, including ties, without implying statistical significance. Additional comparisons are in Appendix G.1.

Setting	m/n	Full	Distortion	Random	Geom.	WFA	DC
$K = 3$, 1 min	24/120	1.2712	**1.2695**	2.2326	1.6518	1.5286	2.5541
$K = 3$, 15 min	24/120	**1.1657**	1.1659	1.5988	1.3081	1.4035	2.1838
$K = 4$, 1 min	42/210	1.1788	**1.1786**	3.4129	1.1895	1.6160	3.5168
$K = 4$, 15 min	42/210	**1.1524**	1.2043	1.5396	2.9478	1.7067	2.4071
$K = 5$, 1 min	50/252	1.0502	**1.0499**	3.1009	1.0706	1.7300	4.9765
$K = 5$, 15 min	50/252	1.3810	**1.2737**	2.2557	1.5288	1.9049	2.7607
$K = 6$, 1 min	42/210	**1.0548**	**1.0548**	1.3530	1.0764	1.6367	5.8050
$K = 6$, 15 min	42/210	**1.3252**	1.3349	1.6344	1.4776	1.9908	2.9673

Exact-value diagnostic. To examine compression with prediction error removed, we freeze the training-selected landmarks and replace their predictions with exact evaluation-day dual values, keeping the reconstruction and decision rule unchanged. This diagnostic uses future information. Oracle Full achieves OPT in every setting, whereas compressed exact values can still incur excess cost. With approximately 20% of values retained, Oracle Distortion's mean ratios range from 1.0005 to 1.0017 across the four one-minute settings. At 15 minutes, the ratio reaches 1.0854 for $K = 6$, compared with 1.0014 at one minute. Thus landmark reconstruction can introduce decision loss even when its inputs are exact, and this loss varies across the evaluated settings. All recorded per-date theoretical cost bounds hold, although they are generally loose (Appendix G.3).

Compression across predictors. We further evaluate the same landmark sets with static, mean, calendar-MLP, and causal-GRU predictors for $K = 4$ at 15 minutes. At a 20% prediction fraction, Distortion achieves lower mean daily ALG/OPT than Random and Geometric for all four predictors, extending the selection advantage beyond the default mean predictor. For the GRU, retaining 42 of 210 predicted dual values yields a ratio of 1.1630, compared with 1.4633 for Random, 2.8774 for

Geometric, and 1.1378 for Full. Increasing the prediction fraction to 30% brings Distortion to 1.1460 (Figure 2(c)). These results show that the same offline landmark sets support effective compression across the tested predictor families, including a recurrent predictor that uses observed request history (Appendix G.2).

Complementing the main experiments, we directly evaluate the learning procedure in Section 5 on synthetic instances by minimizing the joint empirical objective over landmark sets and fixed prediction tables. Results and checks of the theoretical cost bounds appear in Appendix G.5.

7 Related Work

Online MTS and Related Online Problems. MTS was introduced by Borodin et al. (1992), who established the tight deterministic competitive ratio of $2n - 1$. For randomized algorithms, Coester & Lee (2022) give an $O(\log^2 n)$ upper bound for every n-point metric, with a matching worst-case lower bound due to Bubeck et al. (2023). Learning-augmented algorithms use predictions of events or actions to improve online decisions (Mitzenmacher & Vassilvitskii, 2022). For arbitrary MTS, Antoniadis et al. (2023b) combine action predictions with classical competitive algorithms to achieve consistency, smoothness, and robustness, with improved guarantees for caching and online matching on the line. Related approaches use predictions of request arrival times and locations for online TSP and dial-a-ride (Hu et al., 2026), next-arrival times for caching (Wei, 2020), and distributions over the unknown horizon for ski rental (Kang et al., 2026).

Primal–Dual Framework and Predictions. Bamas et al. (2020) introduce a prediction-guided primal–dual framework for online covering, extended to convex covering by Grigorescu et al. (2025). These methods seek consistency with accurate predictions while preserving robustness. Further developments address linear and semidefinite covering programs (Grigorescu et al., 2022), bounded allocation and ad-auctions (Kevi & Nguyen, 2023a), nonlinear packing (Nguyen & Dürr, 2021), and nonlinear covering (Kevi & Nguyen, 2023b). Coester et al. (2026) predict optimal LP dual variables to guide online decisions. For MTS, they interpret the dual as a time-reversed work function and use an A^*-style rule, obtaining $\left(1 + \frac{\eta}{\mathrm{OPT}}\right)$-competitiveness, where η is the sum over rounds of the spans of the Bellman residuals of the predicted values. We use the same value-greedy rule with values reconstructed from landmark predictions.

Compressed Predictions. Prior work studies prediction budgets through fewer oracle calls, fewer bits per prediction, and predictions for selected variables. Im et al. (2022) reduce the number of next-arrival-time predictions used for caching. For MTS, Sadek & Eliáš (2024) query the oracle only every a steps. Shin & Vajanopath (2026) use well-separated queries and virtual predictions for online metric matching. Antoniadis et al. (2023a) provide one bit per request for paging, using discard or phase predictions. In offline approximation algorithms for dense NP-hard problems, Bampis et al. (2024) use one-bit predictions for a logarithmic number of sampled variables. Daneshvaramoli et al. (2025) use a single value or interval estimating the critical value in online knapsack. We study predictions of canonical dual values at a fixed set of landmarks while retaining the full action space. Our excess-cost bounds depend on the landmark geometry and the accuracy of the landmark predictions.

8 Conclusion

We showed how landmark-compressed value predictions reduce prediction dimension in MTS while controlling excess cost. Our bounds link this tradeoff to metric coverage and prediction error, with matching worst-case dependence on the covering radius for fixed-landmark access in the sparse finite-line regime. On two states with one landmark, the Raw/Relative separation shows that retaining absolute cost information can change worst-case expected excess from linear in T to horizon-independent. Beyond coverage, reconstruction distortion captures how well landmarks represent episode values and guides the joint learning of landmarks and fixed prediction tables. Experiments on a Citi Bike-derived K-server benchmark show that distortion-based selection can preserve much of Full's observed performance using a small fraction of predicted values. Natural next steps include adaptive landmark queries and learning guarantees for contextual predictors.

AI USE STATEMENT

We used generative AI tools to assist with the derivation of the learning-theoretic results, code implementation, data processing, and language editing of the manuscript. The authors manually reviewed the AI-assisted work and take responsibility for the final content of the paper, including its theoretical claims, experimental results, and accompanying artifacts.

ETHICS STATEMENT

This work studies online algorithms using synthetic instances and publicly available Citi Bike data. We do not identify any specific ethical concerns arising from this research.

REPRODUCIBILITY STATEMENT

Proofs are provided in Appendices A to E, and experimental protocols in Appendix F. The accompanying `mts-landmark-release.zip` contains source code, experiment configurations, environment specifications, and archived reference results. Its `README.md` and `docs/REPRODUCING.md` describe setup, data preparation, and reproduction.

REFERENCES

Antonios Antoniadis, Joan Boyar, Marek Eliáš, Lene Monrad Favrholdt, Ruben Hoeksma, Kim S. Larsen, Adam Polak, and Bertrand Simon. Paging with succinct predictions. In *International Conference on Machine Learning (ICML)*, pp. 952–968, 2023a.

Antonios Antoniadis, Christian Coester, Marek Eliáš, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. *ACM Transactions on Algorithms*, 19(2):19:1–19:34, 2023b.

Étienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 20083–20094, 2020.

Evripidis Bampis, Bruno Escoffier, and Michalis Xefteris. Parsimonious learning-augmented approximations for dense instances of NP-hard problems. In *International Conference on Machine Learning (ICML)*, pp. 2700–2714, 2024.

Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

Allan Borodin, Nathan Linial, and Michael E. Saks. An optimal on-line algorithm for metrical task system. *Journal of the ACM*, 39(4):745–763, 1992.

Sébastien Bubeck, Christian Coester, and Yuval Rabani. The randomized k-server conjecture is false! In *ACM Symposium on Theory of Computing (STOC)*, pp. 581–594, 2023.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

M. Chrobak, H. Karloff, T. Payne, and S. Vishwanathan. New results on server problems. *SIAM Journal on Discrete Mathematics*, 4(2):172–181, 1991.

Citi Bike. System data: Citi Bike trip histories. `https://citibikenyc.com/system-data`, 2026. Trip-history archives for 2023–2025; accessed September 2026.

Christian Coester and James R. Lee. Pure entropic regularization for metrical task systems. *Theory of Computing*, 18(23):1–24, 2022.

Christian Coester, Alexa Tudose, and Alexander Turoczy. Learning-augmented online minimization with dual predictions. arXiv preprint arXiv:2606.05380, 2026.

Mohammadreza Daneshvaramoli, Helia Karisani, Adam Lechowicz, Bo Sun, Cameron Musco, and Mohammad Hajiesmaili. Near-optimal consistency-robustness trade-offs for learning-augmented online knapsack problems. In *International Conference on Machine Learning (ICML)*, pp. 12459–12489, 2025.

Elena Grigorescu, Young-San Lin, Sandeep Silwal, Maoyuan Song, and Samson Zhou. Learning-augmented algorithms for online linear and semidefinite programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 38643–38654, 2022.

Elena Grigorescu, Young-San Lin, and Maoyuan Song. Learning-augmented algorithms for online concave packing and convex covering problems. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1072–1080, 2025.

Hsiao-Yu Hu, Hao-Ting Wei, Meng-Hsi Li, Kai-Min Chung, and Chung-Shou Liao. Online TSP and online dial-a-ride with predictions. *INFORMS Journal on Computing*, 38(1):189–206, 2026.

Sungjin Im, Ravi Kumar, Aditya Petety, and Manish Purohit. Parsimonious learning-augmented caching. In *International Conference on Machine Learning (ICML)*, pp. 9588–9601, 2022.

Bosun Kang, Hyejun Park, and Chenglin Fan. Learning-augmented ski rental with discrete distribution: A Bayesian approach. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 36671–36678, 2026.

Enikő Kevi and Kim Thang Nguyen. Primal-dual algorithms with predictions for online bounded allocation and ad-auctions problems. In *International Conference on Algorithmic Learning Theory (ALT)*, pp. 891–908, 2023a.

Enikő Kevi and Kim Thang Nguyen. Online primal-dual algorithm with predictions for non-linear covering problems. HAL preprint hal-04361128, 2023b.

Elias Koutsoupias and Christos H. Papadimitriou. On the k-server conjecture. *Journal of the ACM*, 42(5):971–983, 1995.

Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. In *International Conference on Machine Learning (ICML)*, pp. 3296–3305, 2018.

E. J. McShane. Extension of range of functions. *Bulletin of the American Mathematical Society*, 40 (12):837–842, 1934.

Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *Communications of the ACM*, 65(7):33–35, 2022.

Kim Thang Nguyen and Christoph Dürr. Online primal-dual algorithms with predictions for packing problems. arXiv preprint arXiv:2110.00391, 2021.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ML predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9684–9693, 2018.

Karim Abdel Sadek and Marek Eliáš. Algorithms for caching and MTS with reduced number of predictions. In *International Conference on Learning Representations (ICLR)*, 2024.

Yongho Shin and Phanu Vajanopath. Parsimonious learning-augmented online metric matching. arXiv preprint arXiv:2605.26886, 2026.

Alexander Wei. Better and simpler learning-augmented online caching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pp. 60:1–60:17, 2020.

A PRELIMINARIES AND CONVENTIONS

This appendix supplies the Bellman and metric facts used in Sections 2 to 4. Throughout, $\tau = T - 1$, $D = \max_{x,y \in M} d(x, y)$, and all service costs are finite and nonnegative. A value w_t is the optimal remaining cost *after* task t, so $w_T = 0$ and $w_0(s_0) = \mathrm{OPT}$. Landmarks are nonempty and chosen independently of the test episode.

A.1 CANONICAL VALUES AND BELLMAN OPERATORS

Lemma 6 (Bellman Lipschitzness and interval preservation)**.** *For every finite function $f : M \to \mathbb{R}$, $B_c f$ is 1-Lipschitz. The operator is monotone and translation-equivariant. In particular, if $\alpha \le f - g \le \beta$ pointwise, then*

$$\alpha \le B_c f - B_c g \le \beta, \qquad \|B_c f - B_c g\|_{\mathrm{sp}} \le \|f - g\|_{\mathrm{sp}}. \tag{15}$$

Every canonical value in (1) *is 1-Lipschitz.*

Proof. If b minimizes the expression defining $(B_c f)(y)$, the triangle inequality gives

$$(B_c f)(x) \le d(x, b) + c(b) + f(b) \le d(x, y) + (B_c f)(y).$$

Interchanging x, y proves Lipschitzness. Minimizing pointwise inequalities proves monotonicity, and pulling a constant out of a minimum gives $B_c(f + a) = B_c f + a$. Apply these properties to $g + \alpha \le f \le g + \beta$. The final assertion follows from the Bellman recursion and $w_T = 0$. □

The dual interpretation is given in Section 2. Our arguments use the canonical Bellman values, rather than an arbitrary optimal dual solution.

A.2 SPAN ERRORS AND BELLMAN GAPS

For a vector z on a finite nonempty set, let $m_z = \min z$ and $M_z = \max z$. Every constant a satisfies $\max |z - a| \ge (M_z - m_z)/2$, and equality holds at $a = (M_z + m_z)/2$. This proves the identity in (3). The span is invariant under constant shifts and satisfies the triangle inequality.

Lemma 7 (Pathwise gap identity and value-span bound)**.** *For any path starting at s_0, define*

$$\Delta_t = d(s_{t-1}, s_t) + c_t(s_t) + w_t(s_t) - w_{t-1}(s_{t-1}). \tag{16}$$

Then $\Delta_t \ge 0$ and

$$\mathrm{ALG} - \mathrm{OPT} = \sum_{t=1}^{T} \Delta_t. \tag{17}$$

If the path minimizes the predicted Bellman objective at every time and $\widehat{w}_T = 0$, then

$$\mathrm{ALG} - \mathrm{OPT} \le \sum_{t=1}^{\tau} \|\widehat{w}_t - w_t\|_{\mathrm{sp}}. \tag{18}$$

Proof. Nonnegativity follows from Bellman optimality. In the sum of (16), the value terms telescope to $w_T(s_T) - w_0(s_0) = -\mathrm{OPT}$. For the second assertion, let $x_t^\star$ be a Bellman minimizer from the algorithm's actual state s_{t-1}, and set $e_t = \widehat{w}_t - w_t$. Comparing the chosen action to $x_t^\star$ in the predicted objective yields $\Delta_t \le e_t(x_t^\star) - e_t(s_t) \le \|e_t\|_{\mathrm{sp}}$. At $t = T$, the prediction is exact, and hence $\Delta_T = 0$. □

The terminal prediction is the zero function on *all* of M. It is not $E_L 0 = d(\cdot, L)$, which is generally positive away from landmarks. No prediction at time zero is required by the greedy policy.

A.3 Covering radii and prediction dimension

Lemma 8 (Exact covering radius on a finite line)**.** *For $M = \{0, \dots, n-1\}$ with unit-spaced distance and $1 \le m \le n$,*

$$r_m(M) = \left\lceil \frac{n-m}{2m} \right\rceil. \tag{19}$$

Proof. All distances are integers, so a radius-r ball covers at most $2r+1$ states. A cover with at most m balls therefore requires $m(2r+1) \ge n$. Conversely, if this inequality holds, split the line into at most m consecutive blocks of length at most $2r+1$ and choose a middle state in each block. Each block has radius at most r. □

If $\mathcal{N}_M(r) = \min\{|L| : r(L) \le r\}$, then Theorem 2 gives excess at most $2\tau r$ with $\mathcal{N}_M(r)$ exact landmark values per predicted round. Translation equivariance of E_L means that only $m-1$ anchored relative values per round, or $\tau(m-1)$ over an episode, are needed by the greedy policy. This counts scalar values; landmark identities and numerical precision are separate resources. In particular, it is not a bound for arbitrary information encoded into real numbers. Formula (19) also explains why r_m cannot be replaced by a uniform $\Theta(n/m)$ statement near $m = n$, where $r_n = 0$.

B Upper-Bound Proofs and Refined Certificates

We prove the claims in Section 3 and the value-dependent bound used by the learning objective in Section 5.

B.1 Reconstruction and the global guarantee

Proof of Lemma 1. The envelope $E_L v$ is monotone, translation-equivariant, and 1-Lipschitz, by the same minimum-and-triangle-inequality argument as in Lemma 6. If v is 1-Lipschitz on L, then $(E_L v)(\ell) = v(\ell)$ for every landmark: the term indexed by ℓ gives one inequality, and Lipschitzness gives the other. Arbitrary noisy vectors need not be interpolated at their own landmarks.

For the canonical target, $w_t(\ell) + d(x, \ell) \ge w_t(x)$ for all ℓ. Choosing a nearest landmark ℓ_x also gives

$$E_L(w_t|_L)(x) \le w_t(\ell_x) + d(x, \ell_x) \le w_t(x) + 2d(x, L).$$

This proves (6). Put $a_t = \min_{\ell \in L}(v_t(\ell) - w_t(\ell))$ and subtract a_t from every prediction. The landmark errors then lie in $[0, 2\delta_t]$, so monotonicity gives the pointwise bound

$$0 \le E_L(v_t - a_t)(x) - w_t(x) \le 2d(x, L) + 2\delta_t. \tag{20}$$

Taking the span proves (7). The subtraction is only for analysis and does not require the algorithm to know w_t. □

Proof of Theorem 2. Apply Lemma 7 to the predicted functions $\widehat{w}_t = E_L v_t$ for $t < T$ and $\widehat{w}_T = 0$. Summing (7) over the τ predicted rounds gives (8). The comparison uses optimal successors from the algorithm's own states, so it holds for every realized episode and prediction sequence. □

Both $E_L v_t$ and w_t are 1-Lipschitz. Their difference has span at most $2D$, even if the landmark predictions are inconsistent or arbitrarily large. Each summand of the global bound can consequently be replaced by $\min\{2D, 2r(L) + 2\delta_t\}$. This is an additive diameter bound; it does not imply a bounded competitive ratio when OPT is small.

B.2 A local successor-radius guarantee

Define the true Bellman-optimal successor set and its distance to L by

$$\begin{aligned} A_t^\star(a) &= \arg\min_{x \in M} \{d(a, x) + c_t(x) + w_t(x)\}, \\ \rho_t^\star(a; L) &= \min_{x \in A_t^\star(a)} d(x, L). \end{aligned} \tag{21}$$

Theorem 9 (Local successor-radius bound)**.** *The policy in* (5) *satisfies*

$$\mathrm{ALG} - \mathrm{OPT} \le 2 \sum_{t=1}^{\tau} \big[\rho_t^\star(s_{t-1}; L) + \delta_t\big]. \tag{22}$$

Proof. Normalize errors as in (20) and write $e'_t = E_L(v_t - a_t) - w_t \ge 0$. Choose $x_t^\star \in A_t^\star(s_{t-1})$ with smallest distance to L. The action comparison in Lemma 7 gives

$$\Delta_t \le e'_t(x_t^\star) - e'_t(s_t) \le e'_t(x_t^\star) \le 2\rho_t^\star(s_{t-1}; L) + 2\delta_t.$$

Sum over $t < T$ and use $\Delta_T = 0$. □

Here s_{t-1} depends on the candidate landmark set and table. Clustering states on a fixed offline optimal path does not by itself control this certificate. Validation in Appendix E.5 instead evaluates each candidate's own trajectory.

B.3 Value-dependent reconstruction distortion

For an episode X, define

$$\kappa_t(L; X) = \max_{x \in M} \min_{\ell \in L} \{d(x, \ell) + w_t^X(\ell) - w_t^X(x)\}. \tag{23}$$

The exact reconstruction error is nonnegative and vanishes at every landmark. Its minimum is therefore zero, and

$$\kappa_t(L; X) = \|E_L(w_t^X|_L) - w_t^X\|_{\mathrm{sp}} \le 2r(L). \tag{24}$$

Theorem 10 (Distortion-plus-prediction bound)**.** *For every episode and arbitrary landmark predictions,*

$$\mathrm{ALG} - \mathrm{OPT} \le \sum_{t=1}^{\tau} \left[\kappa_t(L; X) + \|v_t - w_t^X|_L\|_{\mathrm{sp}}\right]. \tag{25}$$

Proof. Let a_t and b_t be the minimum and maximum landmark prediction errors. Monotonicity and translation equivariance imply

$$a_t \le E_L v_t - E_L(w_t^X|_L) \le b_t.$$

The span of this difference is at most $b_t - a_t = 2\delta_t$. The span triangle inequality and (24) give $\|E_L v_t - w_t^X\|_{\mathrm{sp}} \le \kappa_t(L; X) + 2\delta_t$. Apply Lemma 7. □

Unlike (22), κ_t is independent of the online trajectory. This is why it can be used in (14). It can be much smaller than $2r(L)$: if $w_t(x) = d(x, a)$ and $L = \{a\}$, the reconstruction is exact and $\kappa_t = 0$, even when $r(L) = D$. All three certificates can also be combined after an episode:

$$\mathrm{ALG} - \mathrm{OPT} \le \sum_{t=1}^{\tau} \min\left\{2D,\ \kappa_t(L; X) + 2\delta_t,\ 2\rho_t^\star(s_{t-1}; L) + 2\delta_t\right\}. \tag{26}$$

Computing these certificates requires offline values; taking online actions does not.

B.4 Comparison with Bellman-residual analysis

To distinguish the direct analysis from a residual bound, let $\widehat{w}_0$ be an auxiliary function and $\widehat{w}_T = 0$, and set

$$h_t = B_{c_t}\widehat{w}_t - \widehat{w}_{t-1}, \qquad \eta_t = \|h_t\|_{\mathrm{sp}}.$$

On the greedy trajectory, the predicted Bellman equality gives $\mathrm{ALG} = \widehat{w}_0(s_0) + \sum_t h_t(s_{t-1})$. Along an offline optimal trajectory, the corresponding inequality gives $\mathrm{OPT} \ge \widehat{w}_0(s_0) + \sum_t \min_x h_t(x)$. Thus $\mathrm{ALG} - \mathrm{OPT} \le \sum_t \eta_t$. Writing $e_t = \widehat{w}_t - w_t$, interval preservation yields

$$\eta_t \le \|e_t\|_{\mathrm{sp}} + \|e_{t-1}\|_{\mathrm{sp}}. \tag{27}$$

For two adjacent exact envelopes, $0 \le e_t, e_{t-1} \le 2r(L)$, and hence $\eta_t \le 4r(L)$. Setting $\widehat{w}_0 = w_0$ for analysis and using $e_T = 0$ yields the total estimate $4\tau r(L) + 4\sum_{t<T} \delta_t$. The direct proof of Theorem 2 pays each current value error once.

Proposition 11 (Tightness of the one-step residual estimate)**.** *The constant* 4 *in the exact-envelope bound* $\eta_t \le 4r(L)$ *cannot be improved in general, even for canonical future values.*

Proof. On the four-state line $M = \{0,1,2,3\}$ take $L = \{0,3\}$, so $r(L) = 1$, and let

$$w_t = (0,1,0,1), \qquad c_t = (2,0,0,2),$$
$$w_{t-1} = B_{c_t} w_t = (2,1,0,1), \quad E_L(w_t|_L) = (0,1,2,1).$$

The preceding envelope is $(2,3,2,1)$, whereas $B_{c_t} E_L(w_t|_L) = (2,1,2,3)$. Their difference is $(0,-2,0,2)$, of span 4. The nonnegative 1-Lipschitz function w_t is realized by a final task $c_{t+1} = w_t$, since $B_{w_t} 0 = w_t$. Thus these are canonical values of a legal episode. □

This example concerns the residual estimate. It does not prove that the coefficient 2 of every algorithmic guarantee is optimal. The matching order lower bound uses the assumptions on fixed landmarks in Theorem 3.

C Full Proof of the Lower Bound for Fixed Landmarks

This appendix completes the construction in Section 4, including finite reset penalties, whole-table indistinguishability, and the quantifiers for randomized landmark selection.

C.1 The gadget and legal service costs

As in Section 4, let $D = n-1$, $h = \lfloor D/(8m) \rfloor$, and $o = s_0$. Choose the baseline $C = D + h$ and the finite penalty $H = 4D$. For $i = 0, \dots, 2m-1$, set

$$p_i = 4hi, \quad I_i = (p_i, p_i + 4h), \quad a_i = p_i + h, \quad z_i = p_i + 2h, \quad b_i = p_i + 3h.$$

Intervals are intersected with the discrete state space when used in set membership. They are disjoint, and all their endpoints lie in M because $8mh \le D$. The assumption $D \ge 16m$ ensures $h \ge 2$.

For one interval, suppress i and define

$$\phi(x) = (h - |x-a|)_+ - (h - |x-b|)_+, \qquad f_\sigma(x) = C - \sigma\phi(x), \quad \sigma \in \{-1,1\}.$$

The two triangular supports have disjoint interiors and slopes of magnitude at most one. Hence f_σ is 1-Lipschitz, $f_\sigma = C$ outside I, and $\{f_\sigma(a), f_\sigma(b)\} = \{C-h, C+h\}$. The four stages described in Section 4 are realized by the following enter, decide, reveal, and reset tasks:

$$c^{\mathrm{in}}(x) = \begin{cases} 0, & x = z, \\ H, & x \ne z, \end{cases} \qquad c^{\mathrm{dec}}(x) = \begin{cases} 0, & x \notin I \text{ or } x \in \{a,b\}, \\ H, & \text{otherwise}, \end{cases}$$
$$c^{\mathrm{rev}}_\sigma(x) = f_\sigma(x) - d(x,o), \quad c^{\mathrm{out}}(x) = \begin{cases} 0, & x = o, \\ H, & x \ne o. \end{cases}$$

In particular, $c^{\mathrm{rev}}_\sigma(x) = f_\sigma(x) - d(x,o) \ge D - d(x,o) \ge 0$. Thus all four tasks have finite, nonnegative service costs.

Why a fixed finite penalty forces a reset. For any 1-Lipschitz continuation V, let a reset task cost zero at q and H elsewhere. From any incoming state x, the score of a non-target state y minus that of q is at least

$$H + d(x,y) - d(x,q) + V(y) - V(q) \ge H - 2D > 0. \tag{28}$$

The reset is therefore uniquely Bellman-optimal. The bound does not depend on the number of later gadgets or their accumulated cost.

C.2 Canonical values throughout the concatenation

Let $g_\sigma = B_{c^{\mathrm{dec}}} f_\sigma$. At z, the favorable candidate costs $h + C - h = C$; the unfavorable one costs $C+2h$; an exterior state costs at least $C+2h$; and any other interior state costs at least $H+C-h > C$.

Thus $g_\sigma(z) = C$. Nonnegative service costs and Lipschitzness imply $g_\sigma \geq B_0 f_\sigma = f_\sigma$. At an exterior state, staying costs exactly C, so

$$g_\sigma(z) = C, \qquad g_\sigma(x) = C \quad (x \notin I). \tag{29}$$

Concatenate $N = \lfloor T/4 \rfloor$ gadgets with interval indices $i_1, \ldots, i_N$ and signs $\sigma_1, \ldots, \sigma_N$, and append $T - 4N$ zero tasks. Define

$$K_j = \sum_{q=j+1}^{N} \big[d(o, z_{i_q}) + C \big]. \tag{30}$$

These constants depend on interval choices but on no hidden sign. Backward induction gives

$$\begin{aligned} w_{4j-4}(x) &= d(x, z_{i_j}) + C + K_j, \\ w_{4j-3}(x) &= g_{\sigma_j}(x) + K_j, \\ w_{4j-2}(x) &= f_{\sigma_j}(x) + K_j, \\ w_{4j-1}(x) &= d(x, o) + K_j. \end{aligned} \tag{31}$$

Here $f_{\sigma_j}, g_{\sigma_j}$ refer to interval i_j. To see the induction, the zero-task suffix has value zero. At a return task, (28) forces o, whose subsequent value is K_j. The reveal task then gives

$$\min_y \{ d(x, y) + f_{\sigma_j}(y) - d(y, o) + d(y, o) + K_j \} = f_{\sigma_j}(x) + K_j,$$

using $B_0 f_{\sigma_j} = f_{\sigma_j}$. The decide task gives $g_{\sigma_j} + K_j$, and the enter task resets to z_{i_j} with continuation $C + K_j$ by (29). Consequently,

$$\mathrm{OPT} = \sum_{j=1}^{N} [d(o, z_{i_j}) + C], \tag{32}$$

also independently of every sign.

C.3 Indistinguishability of the entire Raw table

Fix L and an unhit interval $L \cap I_{i_j} = \varnothing$. At each landmark, f_{σ_j} and g_{σ_j} both equal C. Equations (30)–(32) therefore show that flipping σ_j leaves *every* revealed dual value unchanged, at every time including time zero. In particular, no earlier absolute value reveals the sign through its continuation cost.

The tasks through the decide stage of gadget j do not depend on σ_j either. Conditional on the interval choices, the other signs, and the algorithm's randomness, the two choices of σ_j thus induce identical observations and identical actions until that gadget's reveal stage. With a fair sign, it remains a fair hidden bit at the decision round, even when the full Raw table was given in advance.

C.4 Loss on an unhit interval

Use the Bellman gaps in Lemma 7. If the enter action is not z, then (28) gives $\Delta_{\mathrm{in}} \geq H - 2D \geq h$. Otherwise the decision begins at z, where the Bellman value is $C + K_j$. The favorable candidate has gap zero, the unfavorable candidate has gap $2h$, an exterior state has gap at least $2h$, and any other interior state has gap at least $H - h$. Because the sign is hidden, either candidate has conditional expected gap h, and every other action has at least this much. This also holds for randomized mixtures of actions. Therefore

$$\mathbb{E}[\Delta_{\mathrm{in}} + \Delta_{\mathrm{dec}} \mid L \cap I_{i_j} = \varnothing] \geq h. \tag{33}$$

Later gaps are nonnegative and cannot cancel this loss.

C.5 Oblivious inputs and the final constant

Choose every interval index independently and uniformly from $\{0, \ldots, 2m - 1\}$, and choose all signs independently and uniformly. This distribution fixes the full episode without observing any

realized algorithmic random bits. Any realized landmark set with $|L| \le m$ hits at most m of the $2m$ disjoint intervals. As L is chosen before test information is observed,

$$\Pr[L \cap I_{i_j} = \varnothing \mid L] \ge \tfrac{1}{2}.$$

Combining this with (33) and summing gaps yields $\mathbb{E}[\mathrm{ALG} - \mathrm{OPT}] \ge Nh/2$, with expectation over both input and algorithm randomness. Averaging over the finite input support fixes a deterministic episode whose expected excess over algorithm randomness is at least $Nh/2$.

Finally, $D \ge 16m$ implies $h = \lfloor D/(8m) \rfloor \ge D/(16m)$. By Lemma 8, $r_m(M) \le D/m$ in this regime, so $h \ge r_m(M)/16$. Therefore the selected episode satisfies

$$\mathbb{E}[\mathrm{ALG}] - \mathrm{OPT} \ge \frac{1}{32} \left\lfloor \frac{T}{4} \right\rfloor r_m(M),$$

which proves Theorem 3. The zero-task padding leaves all preceding canonical values unchanged and adds no negative gap.

This proof requires the landmark set to be fixed before the episode. Adaptively querying the future value at a_i and at a state outside I_i reveals a value difference $-\sigma h$ at the decision stage. It therefore does not establish an adaptive-query lower bound. The nearly-full Raw regime is addressed separately in Appendix D.

D Absolute and Relative Prediction Interfaces

This appendix proves the interface separation in Theorem 4 and explains the sparse-regime restriction in Theorem 3. The guarantees here allow policies other than the value-greedy rule.

D.1 A Raw budget policy with one unreported state value

Theorem 12 (One unreported state value under Raw access)**.** *Suppose $M \setminus L = \{q\}$ and $L \ne \varnothing$. With round-by-round access to exact absolute landmark values, a deterministic policy satisfies*

$$\mathrm{ALG} \le \mathrm{OPT} + 2r(L) \tag{34}$$

on every episode, without being supplied OPT.

Policy. For $T = 1$, minimize the observed movement-plus-service cost directly. For $T \ge 2$, after receiving c_1 and $w_1|_L$, set

$$U = \min_{x \in M} \{d(s_0, x) + c_1(x) + E_L(w_1|_L)(x)\}. \tag{35}$$

Let P_{t-1} be the amount already paid and $B_{t-1} = U - P_{t-1}$. At time t, select a landmark satisfying

$$d(s_{t-1}, \ell) + c_t(\ell) + w_t(\ell) \le B_{t-1}, \tag{36}$$

if one exists, using a fixed tie rule. Otherwise select the unique state q outside L. At $t = T$, the required values are the known zeros.

Proof of Theorem 12. Exact reconstruction gives $w_1 \le E_L(w_1|_L) \le w_1 + 2r(L)$, and hence $\mathrm{OPT} \le U \le \mathrm{OPT} + 2r(L)$. Maintain the invariant

$$B_{t-1} \ge w_{t-1}(s_{t-1}). \tag{37}$$

It holds initially. If a feasible landmark is chosen, subtracting its immediate cost from (36) proves the next instance of the invariant. If no landmark is feasible, Bellman optimality and (37) guarantee that some state has movement-plus-service-plus-continuation cost at most B_{t-1}. That state must be q, so choosing it preserves the invariant without observing $w_t(q)$. At the end, $U - \mathrm{ALG} = B_T \ge w_T(s_T) = 0$, proving the bound. □

If OPT is additionally revealed, initializing $U = \mathrm{OPT}$ makes the policy optimal. This extra information is not assumed above; it is available, for example, if the Raw interface also reveals $w_0(s_0)$. The policy depends on a common absolute cost scale across times, so independent additive shifts of successive revealed value vectors invalidate its budget comparisons. This is compatible with shift invariance of the different greedy policy in Section 3.

D.2 Proof of the two-state separation

Proof of Theorem 4. The Raw upper bound is Theorem 12. With one landmark, every revealed relative value is zero, regardless of the chosen landmark. Label the initial state 0, the other state 1, and set $H = 4a$. For each independent gadget, choose a fair hidden bit $b \in \{0, 1\}$ and use the following tasks:

task	$c(0)$	$c(1)$
enter	0	H
decide	a	0
reveal	$H\,1\{b = 1\}$	$H\,1\{b = 0\}$
return	0	H

Every reset is Bellman-optimal by (28), with $D = a$. Failure to enter 0 creates gap at least $H - 2a = 2a$. Otherwise, both decision actions have immediate cost a. Writing K for the continuation cost after returning to 0, the future value after the decision at state x is $d(x, b) + d(b, 0) + K$. The Bellman-optimal decision is b, and choosing the other state has gap exactly a. The current task, all past tasks, and the identically zero relative values reveal nothing about the fresh bit. Thus the conditional expected decision gap is at least $a/2$. This argument covers every randomized decision rule.

Repeat $\lfloor T/4 \rfloor$ gadgets and append zero tasks. Sum the nonnegative Bellman gaps and average over the hidden bits to fix an oblivious input satisfying (10). □

The Raw excess bound is independent of T, whereas the Relative interface can force excess linear in T. Thus the landmark budget alone does not characterize the information available to arbitrary history-dependent policies.

E Learning Guarantees and Optimization Details

We supply proofs for Section 5. Training samples are N_{train} independent complete episodes from $\mathcal{D}$; no independence among the tasks of a single episode is assumed. Every fitted table is fixed across test episodes. Set $\tau = T - 1$ and choose an anchor $\ell_0 \in L$ by a deterministic rule.

E.1 Envelope closure and a compact class

Lemma 13 (Closure preserves decisions and contracts target error)**.** *For any vector v on L, let $\bar{v} = (E_L v)|_L$. Then $\bar{v}$ is 1-Lipschitz, $E_L \bar{v} = E_L v$, and, for every 1-Lipschitz target w,*

$$\|\bar{v} - w|_L\|_{\mathrm{sp}} \le \|v - w|_L\|_{\mathrm{sp}}. \tag{38}$$

Proof. The envelope is 1-Lipschitz, and its restriction is therefore 1-Lipschitz. Expanding the repeated envelope gives

$$E_L \bar{v}(x) = \min_{\ell, \ell' \in L} \{v(\ell') + d(\ell, \ell') + d(x, \ell)\}.$$

The triangle inequality bounds this below by $E_L v(x)$. Taking $\ell = \ell'$ gives the reverse inequality. If $a \le v - w|_L \le b$, monotonicity and translation equivariance yield $E_L(w|_L) + a \le E_L v \le E_L(w|_L) + b$. The exact envelope agrees with w on L, so restricting this inequality to L proves (38). □

After closure, subtracting $\bar{v}(\ell_0)$ only translates the extended function. Thus there is no loss of greedy decision rules or increase of span loss in restricting to

$$\mathcal{V}_L = \{v : v_t(\ell_0) = 0, \quad |v_t(\ell) - v_t(\ell')| \le d(\ell, \ell') \text{ for all } t, \ell, \ell'\}. \tag{39}$$

This is a closed bounded polytope with $\tau(m - 1)$ free parameters, each in $[-D, D]$. It is nonempty, and continuous empirical risks attain their minima on it. The anchored true labels $q_t^X(\ell) = w_t^X(\ell) - w_t^X(\ell_0)$ are also 1-Lipschitz. For any u, v on L,

$$\left| \|u - q_t^X\|_{\mathrm{sp}} - \|v - q_t^X\|_{\mathrm{sp}} \right| \le 2\|u - v\|_\infty. \tag{40}$$

If v is feasible, its difference from q_t^X has span at most $2D$. These bounds hold independently of the absolute scale of the service costs.

E.2 Proof of the fixed-table PAC guarantee

Proof of Theorem 5. Fix t and write $\ell_{t,v_t}(X) = \|v_t - q_t^X\|_{\mathrm{sp}}$. Partition $[-D, D]^{m-1}$ into cells of side $a = \varepsilon/16$, choosing a feasible representative in each nonempty cell. The representatives form an a-net of size at most

$$N_\varepsilon \le (1 + 32D/\varepsilon)^{m-1}.$$

For a fixed representative, the N_{train} losses are independent across episodes and lie in $[0, 2D]$. Hoeffding's inequality gives

$$\Pr\left[\left|\frac{1}{N_{\mathrm{train}}}\sum_{i=1}^{N_{\mathrm{train}}} \ell_{t,v_t}(X^{(i)}) - \mathbb{E}_X \ell_{t,v_t}(X)\right| > \frac{\varepsilon}{4}\right] \le 2\exp\left(-\frac{N_{\mathrm{train}}\varepsilon^2}{32D^2}\right). \tag{41}$$

A union bound over all representatives and all τ times has failure probability at most β under (12). On the resulting event, replacing an arbitrary feasible vector by its representative changes empirical and population losses by at most $2a$ each, by (40). Therefore the deviation at each time is at most $\varepsilon/4 + 4a = \varepsilon/2$ uniformly over its feasible vectors. Averaging over time proves

$$\sup_{v \in \mathcal{V}_L} |\widehat{R}_L(v) - R_L(v)| \le \varepsilon/2. \tag{42}$$

For a population minimizer $v^\star$ and any empirical minimizer $\widehat{v}$, this implies

$$R_L(\widehat{v}) \le \widehat{R}_L(\widehat{v}) + \varepsilon/2 \le \widehat{R}_L(v^\star) + \varepsilon/2 \le R_L(v^\star) + \varepsilon.$$

Finally, $\|v_t - q_t^X\|_{\mathrm{sp}} = 2\delta_t$ and Theorem 2 imply (13) after taking expectation over a fresh test episode. □

The expectation over the test episode is conditional on the learned table; the high-probability event is over training episodes. The union bound over times needs no within-episode independence. Because the risk averages over τ times, a total statistical excess target $\varepsilon_{\mathrm{tot}}$ requires setting $\varepsilon = \varepsilon_{\mathrm{tot}}/\tau$, producing a quadratic horizon factor in the sample bound. For $m = 1$, the anchored class contains only the zero table and its span risk is identically zero; compression can still incur nonzero online excess.

E.3 The empirical-risk linear program

Let $q_{it\ell} = q_t^{X^{(i)}}(\ell)$. The LP used in Sections 5 and 6 is

$$\begin{aligned}
\min_{v,u,b} \quad & \frac{1}{N_{\mathrm{train}}\tau}\sum_{i=1}^{N_{\mathrm{train}}}\sum_{t=1}^{\tau}(u_{it} - b_{it}) & \\
\text{subject to} \quad & v_t(\ell) - q_{it\ell} \le u_{it} & \text{for all } i, t, \ell, \\
& b_{it} \le v_t(\ell) - q_{it\ell} & \text{for all } i, t, \ell, \\
& v_t(\ell) - v_t(\ell') \le d(\ell, \ell') & \text{for all } t, \ell, \ell', \\
& v_t(\ell_0) = 0 & \text{for all } t.
\end{aligned} \tag{43}$$

The ordered-pair constraints include both directions of the Lipschitz condition. At an optimum, u_{it} is the largest landmark error and b_{it} the smallest, since otherwise either variable can be tightened to decrease the objective. The objective is thus exactly $\widehat{R}_L(v)$.

Before eliminating anchors, the LP has $\tau m + 2N_{\mathrm{train}}\tau$ variables, $2N_{\mathrm{train}}\tau m + \tau m(m-1)$ displayed inequalities, and τ anchor equalities. Bounds $[-D, D]$ on non-anchor table entries are redundant mathematically and are imposed in the implementation. The variables u, b are unrestricted real numbers. With rational data, the labels and this LP have polynomial encoding size and the fixed-L problem is polynomial-time solvable. Computing all training labels by generic dynamic programming costs $O(N_{\mathrm{train}}Tn^2)$.

Solver and storage conventions are given in Appendix F.9.

E.4 JOINT SELECTION OF LANDMARKS AND TABLES

Let $\widehat{J}$ be the empirical version of (14), with κ_t given by (23).

Theorem 14 (Joint landmark/table generalization)**.** *If*

$$N_{\text{train}} \geq \frac{128D^2}{\varepsilon^2}\left[\log\binom{n}{m} + (m-1)\log\left(1+\frac{32D}{\varepsilon}\right) + \log\frac{2\tau}{\beta}\right], \tag{44}$$

then, with probability at least $1-\beta$*, every output* $(\widehat{L},\widehat{v})$ *whose empirical objective is within* $\gamma \geq 0$ *of the global empirical minimum satisfies*

$$\mathbb{E}_X[\text{ALG}_{\widehat{L},\widehat{v}} - \text{OPT}] \leq \tau\left[\inf_{|L|=m,\, v\in\mathcal{V}_L} J(L,v) + \varepsilon + \gamma\right]. \tag{45}$$

Proof. For each L, every timewise summand in J lies in $[0,4D]$: both κ_t and the landmark error span are at most $2D$. The distortion is independent of v_t, so the summand remains 2-Lipschitz in that vector. Use the same $\varepsilon/16$ net as in Appendix E.2, and union-bound over all $\binom{n}{m}$ landmark sets as well as the τ times. The exponent in (41) becomes $-N_{\text{train}}\varepsilon^2/(128D^2)$. Under (44), $|J-\widehat{J}| \leq \varepsilon/2$ uniformly. For a population minimizer $(L^\star, v^\star)$, empirical suboptimality gives

$$\begin{aligned} J(\widehat{L},\widehat{v}) &\leq \widehat{J}(\widehat{L},\widehat{v}) + \varepsilon/2 \\ &\leq \widehat{J}(L^\star,v^\star) + \gamma + \varepsilon/2 \leq J(L^\star,v^\star) + \gamma + \varepsilon. \end{aligned}$$

Use Theorem 10 to convert risk to excess. □

For each L, minimizing $\widehat{J}$ uses the same LP because its distortion term is constant in v. Enumerating all $\binom{n}{m}$ sets therefore realizes global empirical minimization in exact arithmetic. The combinatorial factor is not polynomial for unrestricted m. The pilots enumerate all 36 pairs and solve their LPs numerically; they do not provide exact-arithmetic optimality certificates. A general heuristic requires an explicit optimization-error allowance γ unless its global suboptimality is certified.

E.5 INDEPENDENT VALIDATION OF FROZEN CANDIDATES

Let $\mathcal{A}_1,\ldots,\mathcal{A}_{J_0}$ be value-greedy policies generated and fitted using only training data, and then frozen. For each candidate define its normalized episode excess

$$Z_j(X) = \frac{\text{ALG}_j(X) - \text{OPT}(X)}{\tau}.$$

Select $\widehat{j}$ by minimizing mean Z_j over independent validation episodes.

Theorem 15 (Finite-candidate validation)**.** *If*

$$N_{\text{val}} \geq \frac{8D^2}{\varepsilon^2}\log\frac{2J_0}{\beta}, \tag{46}$$

then, with probability at least $1-\beta$*,* $\mathbb{E}_X Z_{\widehat{j}}(X) \leq \min_{j\leq J_0}\mathbb{E}_X Z_j(X) + \varepsilon$.

Proof. Condition on the training set and all randomness used to construct the candidates. By Lemma 7 and the Lipschitz diameter bound, $0 \leq Z_j \leq 2D$ for every candidate and episode. Hoeffding's inequality gives a failure probability at most $2\exp(-N_{\text{val}}\varepsilon^2/(8D^2))$ for deviation $\varepsilon/2$ for one candidate. Union-bound over J_0 candidates and compare the empirical winner to the population-best candidate, using the deviation bound twice. The conclusion holds for every conditioning and hence also without conditioning. □

The guarantee compares to the best member of the frozen family, not the globally best landmark set. It permits arbitrary training-only candidate generation and needs no global optimization certificate. Repeatedly modifying candidates using validation outcomes would no longer satisfy the stated frozen-family assumption. Test episodes are used only for final evaluation. None of the learning results above asserts a complexity bound for unrestricted contextual predictors.

F EXPERIMENTAL PROTOCOL AND REPRODUCIBILITY

This appendix expands the protocol in Section 6, first for the Citi Bike-derived benchmark and then for the synthetic fixed-table pilots. Additional results appear in Appendix G.

F.1 DEMAND DATA AND PREPROCESSING

The benchmark uses trip start times and coordinates from the official Citi Bike archive (Citi Bike, 2026). The frozen input comprises the 2023 annual archive and twelve monthly archives for each of 2024 and 2025: 25 ZIP files. Across 1,096 unique dates, 125,181,021 raw trip records yield 100,861,883 records after filtering. Table 2 specifies the chronological split. The final refit includes the initial training and validation periods; it is not an additional independent data block.

Table 2: Citi Bike data split. The final refit overlaps the two preceding rows.

Split	Dates	Days	Filtered trips
Initial fit	2023-01-01–2024-09-30	639	55,497,434
Validation	2024-10-01–2024-12-31	92	8,910,606
Final refit	2023-01-01–2024-12-31	731	64,408,040
Evaluation	2025-01-01–2025-12-31	365	36,453,843

The geographic filter is the rectangle with latitude $[40.70, 40.88]$ and longitude $[-74.025, -73.90]$, not an exact borough boundary. The observed latitude range in the initial 639 training days is divided into ten equal-width bins and frozen. Later coordinates outside that fitted range are clipped to the edge bins. Longitude is used only in the rectangle filter; distances are measured in latitude-bin units. There is no road network, inventory, capacity, return-trip, traffic, or truck-loading model.

Each slot requests the bin with the largest trip count, breaking count ties toward the smallest index. Empty slots receive the special label -1; missing date coverage raises an error. Fifteen-minute requests are formed by summing minute/bin counts before taking the most frequent bin, rather than aggregating minute-level modal labels. Archive-local New York wall-clock times give 96 or 1,440 slots per day, with repeated daylight-saving slots merged. Every day starts a new service episode.

Modal aggregation concentrates the requests: bins 1 and 2 account for 90.79% of nonempty 15-minute requests and 75.72% of nonempty one-minute requests across the processed traces. Mean active bins per day are 3.89 and 8.09, respectively; within-day adjacent-repeat rates, including empty labels, are 69.81% and 40.44%. The 2025 traces contain eight empty 15-minute slots and 3,639 empty minute slots. These statistics delimit the benchmark's scope. The finer trace is less concentrated but can be easier to compress, so concentration alone does not explain the results.

F.2 CONFIGURATION METRIC, ONLINE RULE, AND MTS ENCODING

Use K for the number of servers and $m = |L|$ for the landmark budget, consistently with the theory. The report's state count N is denoted by n here; its landmark budget m uses the same notation. A state is a sorted set of distinct bins:

$$\mathcal{C}_K = \{C \subseteq \{0, \dots, 9\} : |C| = K\}, \quad n = \binom{10}{K}, \quad d(C, C') = \sum_{i=1}^{K} |c_i - c'_i|. \tag{47}$$

The MTS states are whole configurations in the induced matching metric. The starting configurations for $K = 3, 4, 5, 6$ are respectively $\{0, 4, 9\}$, $\{0, 3, 6, 9\}$, $\{0, 2, 4, 7, 9\}$, and $\{0, 2, 4, 5, 7, 9\}$; the state counts are $120, 210, 252, 210$.

Using the paper's one-based request convention, let r_t be the request at round t. The exact continuation values and the predicted-value rule are

$$w_T^d(C) = 0, \qquad w_{t-1}^d(C) = \min_{C' : r_t^d \in C'} \{d(C, C') + w_t^d(C')\}, \tag{48}$$

$$C_t \in \arg\min_{C' : r_t \in C'} \{d(C_{t-1}, C') + \widetilde{P}_t(C')\}. \tag{49}$$

Here d as a superscript denotes the date, and $\mathrm{OPT}_d = w_0^d(C_0)$. For empty requests, the implementation uses the identity Bellman operator and leaves the online configuration unchanged. For nonempty requests, several servers may move and all movement is charged. Scores within 10^{-12} of the minimum are tied; smaller movement is preferred, followed by the fixed configuration index. Exact DP uses float64. Oracle Full receives exact evaluation-day values at every state and must attain OPT; fitted predictors never receive these values during decisions.

Finite service costs. The hard feasibility condition admits an equivalent finite-cost MTS encoding for these value-greedy policies and the offline optimum. Let $D_{\mathcal{C}}$ be the configuration diameter, choose any $H > 2D_{\mathcal{C}}$, and set $c_t(C) = 0$ if $r_t \in C$ and $c_t(C) = H$ otherwise. For any 1-Lipschitz continuation f, any infeasible state A, and any feasible state B,

$$d(C, B) + f(B) - [d(C, A) + H + f(A)] \leq 2d(A, B) - H < 0.$$

An infeasible state therefore cannot minimize the finite-cost objective. Backward induction from the terminal zero identifies its canonical values with (48); all evaluated predictions and their extensions are also 1-Lipschitz. For an empty request, set $c_t \equiv 0$: the Bellman operator fixes a 1-Lipschitz vector, and minimum-movement tie-breaking selects the current state. The implementation directly enforces feasibility, avoiding a large numerical penalty.

Classical references. WFA (Koutsoupias & Papadimitriou, 1995) maintains the extended work function for the request history and the fixed initial configuration, then minimizes its work value plus movement over feasible next configurations. DC (Chrobak et al., 1991) moves the closest server for a request outside the occupied interval; between servers, the two neighbors move at equal speed until one reaches the request. DC may use coincident continuous positions, while DP and the prediction policies use discrete K-element subsets. We retain this implementation difference explicitly and use their measured costs, rather than importing a worst-case theorem as an empirical guarantee. All reported dates have positive OPT. Mean daily OPT for $K = 3, 4, 5, 6$ at 15 minutes is respectively $10.1973, 7.3589, 4.4685, 4.0712$; for $K = 4$ at one minute it is 61.2164.

F.3 Predictors, landmark fitting, and prediction access

The mean predictor uses the same table on every evaluation date:

$$P_t^{\mathrm{mean}}(C) = \frac{1}{|\mathcal{D}_{\mathrm{fit}}|} \sum_{d \in \mathcal{D}_{\mathrm{fit}}} w_t^d(C), \qquad \widetilde{P}_t(C) = \min_{\ell \in L} \{P_t(\ell) + d(C, \ell)\}.$$

The average of canonical values is 1-Lipschitz. The extension interpolates the retained predicted dual values, and $L = \mathcal{C}_K$ recovers Full. The terminal prediction is explicitly zero for every method, including compressed ones.

Random uses nested prefixes of a seeded random permutation of all states. Geometric starts at a distance medoid and adds the state farthest from its nearest landmark. Distortion greedily minimizes

$$\widehat{J}_{\mathrm{recon}}(L) = \frac{1}{S} \sum_{j=1}^{S} \left\| E_L(v^j|_L) - v^j \right\|_{\mathrm{sp}}, \qquad S \leq 1024, \tag{50}$$

on sampled training day/slot exact-value vectors, reusing the same rows throughout a greedy run. Each addition minimizes this empirical objective among candidates; there is no global subset optimization or test-cost minimization. The exact-value reconstruction error is nonnegative at every state and zero at landmarks, so its span equals its maximum over states. Distortion seeds 0–4 vary the sampled rows; Random seeds 0–9 vary the permutations. Budgets use nested prefixes within each seed. These seeds do not resample entire training datasets.

Equation (50) selects landmarks using reconstruction distortion, with predictors fitted separately; see Appendix G.6 for its relation to the learning guarantees. Across the temporal predictors, the same original landmark sets are reused within each split, with no predictor-specific or evaluation-specific refitting of those sets.

Landmark identities and the known metric are outside the prediction budget. The prediction budget is m values per slot, reported as a fraction m/n. The implementation computes full prediction vectors before restricting them to landmarks; this budget measures values supplied to the policy, not end-to-end computational, storage, or communication savings.

F.4 Temporal predictors and structural Lipschitz outputs

The temporal extension fixes $K = 4$ and 15-minute requests. Static potential averages the mean table over nonterminal times and holds that configuration potential constant until the terminal zero. The calendar MLP receives six features: sine/cosine pairs for time of day, weekday, and position in the year, accounting for leap-year length. A 32-unit Tanh hidden layer is concatenated with those calendar features before the n-value linear head. The GRU (Cho et al., 2014) has one 32-unit recurrent layer, receiving the same calendar features and an 11-class one-hot request vector (ten bins and the empty label). Its hidden state and calendar features feed the linear head. The MLP has 8,414 parameters and the GRU 13,086; this is not a parameter-matched comparison.

The hidden state resets daily. In the one-based notation of (49), P_t uses only the requests $r_1, \dots, r_t$ already observed before that decision. Equivalently, the implementation's zero-based P_{j+1} follows the update on r_j. Predictions do not depend on the policy's server state, so all compression variants receive the same causally available predictor values. No feature scaler is fitted.

For arbitrary network scores z_t, the deterministic output map is

$$a_t(C) = \min_{C'}\{z_t(C') + d(C, C')\}, \qquad b_t(C) = \max_{C'}\{z_t(C') - d(C, C')\},$$
$$q_t(C) = \tfrac{1}{2}[a_t(C) + b_t(C)], \qquad P_t(C) = q_t(C) - \frac{1}{n}\sum_{C'} q_t(C'), \qquad P_T = 0. \tag{51}$$

Every distance cone is 1-Lipschitz; taking a finite minimum or maximum preserves the common bound, as do averaging and subtracting a constant. Consequently, every parameter setting and history yields spatially 1-Lipschitz values. If z_t is already 1-Lipschitz, both envelopes equal z_t. This is a constraint map, not a claimed Euclidean projection or a Lipschitz guarantee with respect to input histories.

The target is $Y_t^d(C) = w_t^d(C) - n^{-1}\sum_{C'} w_t^d(C')$. Training minimizes mean squared error over dates, nonterminal times, and configurations. AdamW uses learning rate 10^{-3}, weight decay 10^{-4}, 16 complete days per batch, gradient clipping at 1.0, at most 100 epochs, and patience 12. Training is float32; at evaluation, raw scores are converted to float64 before the output map, reconstruction, and rollout. Deterministic algorithms are enabled and TF32 is disabled. Gradient clipping is unrelated to the spatial guarantee. Each network uses training seeds 0, 1, and 2. Validation MSE selects the epoch count, followed by a fresh refit on all 731 fitting days for that count. All selection runs reach the 100-epoch cap; this fixed-budget protocol does not establish convergence to the best possible model.

F.5 Budgets and data sensitivity

The mean-predictor suite uses nominal fractions $\rho \in \{0.05, 0.10, 0.20, 0.30, 0.40, 0.60, 0.80, 1.00\}$ and $m = \max\{1, \mathrm{round}(\rho n)\}$. For $n = 210$, the budgets are $10, 21, 42, 63, 84, 126, 168, 210$; the nominal 5% budget is actually 4.76%. For $K = 5$, the nominal 20% budget is $50/252 = 19.84\%$. The original five settings comprise four server counts at 15 minutes and $K = 4$ at one minute. The supplement adds $K = 3, 5, 6$ at one minute, so all eight settings have validation and annual evaluation runs. The original predictor and landmark settings are frozen; validation records diagnostics rather than supporting a broad hyperparameter search. The temporal extension uses budgets $10, 21, 42, 63, 210$, crossing each of three neural seeds with ten Random or five Distortion seeds.

Budgets count predicted values per slot, not per day. At a fixed m, the one-minute horizon supplies 15 times as many values per episode as the 15-minute horizon. Both traces come from the same raw trips, but count-based modal aggregation changes the requests as well as the horizon and OPT; it is not simple subsampling. Cross-resolution comparisons do not hold daily communication volume or the request distribution fixed.

Seven further runs fix $K = 4$, 15-minute requests, and $m = 42$, using $8, 16, 32, 64, 128, 256, 512$ fitting days. These are nested prefixes of one seed-0 permutation of the 731-day fitting pool; the full-pool run supplies the 731-day reference. The geographic bins and metric still use the initial 639-day calibration, which is not counted on the fitting-day axis. The experiment measures sensitivity along one date-sampling path, not repeated training-set sampling or empirical sample-complexity scaling.

Neural validation scores select the epoch count; final evaluation uses separately refitted networks. Validation and annual evaluation have different fitting pools and seasons.

The first ten evaluation dates (January 1–10, 2025) were used in preliminary development, and additional analyses were developed after inspecting the evaluation year. Excluding these dates is a sensitivity check rather than a fresh holdout evaluation (Appendix G.4).

F.6 Metrics and statistical interpretation

For each date, average all applicable training-seed and landmark-seed combinations before averaging across dates:

$$R_d = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{\mathrm{ALG}_{d,s}}{\mathrm{OPT}_d}, \qquad \bar{R} = \frac{1}{|\mathcal{D}_{\mathrm{eval}}|}\sum_{d} R_d, \qquad \mathrm{Ret}(m) = \frac{B-\bar{R}_m}{B-\bar{R}_{\mathrm{Full}}}, \tag{52}$$

where $B = \min\{\bar{R}_{\mathrm{WFA}}, \bar{R}_{\mathrm{DC}}\}$ and performance retention is defined only when Full improves on B. This is the mean of daily ratios, not a ratio of annual total costs or an estimated worst-case competitive ratio. Each date has equal weight regardless of OPT. Retention above 100% means lower observed cost than Full; negative retention means worse cost than the classical reference. Its denominator depends on the predictor, so cross-predictor comparisons must also use the actual costs.

With B_t denoting the Bellman operator for the observed request, report

$$\eta_d(P) = \sum_{t=1}^{T}\|B_tP_t - P_{t-1}\|_{\mathrm{sp}}, \qquad E_{\mathrm{recon}} = \frac{1}{T}\sum_{t=0}^{T-1}\|\widetilde{P}_t - P_t\|_{\mathrm{sp}}. \tag{53}$$

Full has zero reconstruction error relative to itself, but generally nonzero residual. This reconstruction diagnostic is not the exact-value distortion $\kappa_t(L;X)$ in the theory: it compares against predicted Full values rather than w_t. Centered value RMSE and value-error span are additional temporal diagnostics. For a fixed mean table, reconstruction span is constant across evaluation dates, so a conditional date interval can collapse to a point. These diagnostics need not rank predictors in the same order as online cost.

Original-suite marginal intervals use 2,000 ordinary date-bootstrap replicates. Its paired comparisons additionally use circular seven-day blocks (2,000 replicates, seed 20260919). Temporal headline and paired intervals use seven-day blocks (2,000 replicates, seed 20260921); ordinary date intervals are also preserved in the experiment archive. Negative paired differences favor the left-hand method. Supplementary mean, cross-resolution and oracle intervals use 2,000 circular seven-day block replicates with seed 20260922; ordinary date intervals are also saved. Recomputed control intervals can differ slightly from the original report because their bootstrap seed changes. Tables identify their source protocol, and original observations are not pooled again as new evidence. All intervals condition on the fixed training data and seed set; they do not include uncertainty from independently resampled training datasets. Blocks retain some short-term dependence but do not eliminate seasonal or cross-year shifts. Comparisons are unadjusted for multiplicity; crossing zero is not an equivalence test.

F.7 Frozen-landmark oracle protocol and certificate indexing

The exact-value diagnostic covers all eight server-count/resolution settings, using the training-fitted landmark sets, budgets, numerical precision and tie rules from Appendices F.2 and F.5. Random has ten selection seeds, Distortion five, and Geometric one.

Oracle inputs and interpretation. The oracle replaces predicted dual values at frozen landmarks by their exact evaluation-day values. It uses the same McShane reconstruction, terminal zero, empty-request convention, and value-greedy action rule. Oracle Full must achieve OPT. At fixed L, predicted-compressed minus oracle-compressed cost is the net effect of replacing predicted dual values; it can have either sign. Oracle excess is the decision loss with exact dual values and the chosen reconstruction. These comparisons do not identify independent additive causal sources of prediction and compression loss. In particular, fitted landmark sets, request distributions, horizons, and OPT all differ between resolutions.

Distortion and certificate indexing. The report's zero-based V_t^*, defined before request r_t, is the paper's w_t, defined after the first t requests. For a fixed date and landmark set, put $\widetilde{w}_t = E_L(w_t|_L)$ for $t < T$ and override $\widetilde{w}_T = 0$. Define

$$e_t = \|\widetilde{w}_t - w_t\|_{\mathrm{sp}}, \qquad e_T = 0, \qquad S_0 = \sum_{t=0}^{T-1} e_t, \qquad \bar{e} = S_0/T. \tag{54}$$

For the decision times $1 \le t < T$, e_t is exactly the paper's $\kappa_t(L; X)$ in (23). The report's κ_Σ is S_0, including e_0; it is not the decision-time sum $\sum_{t=1}^{T-1} \kappa_t = S_0 - e_0$. Its reported "mean distortion" is $\bar{e}$, distinct from the predictor-relative reconstruction error in (53). Across resolutions, S_0 additionally depends on the number of slots.

Let $\mathcal{A} = \{t \in \{1, \dots, T\} : r_t \neq -1\}$ use the paper's one-based request convention. The archived bounds become

$$0 \le \mathrm{ALG}_{\mathrm{or}} - \mathrm{OPT} \le \eta(\widetilde{w}) \le \underbrace{e_0 + 2\sum_{t=1}^{T-1} e_t}_{B_{\mathrm{adj}} = 2S_0 - e_0}, \tag{55}$$

$$\mathrm{ALG}_{\mathrm{or}} - \mathrm{OPT} \le \underbrace{\sum_{t \in \mathcal{A}} e_t}_{B_{\mathrm{act}}} \le S_0 - e_0. \tag{56}$$

Here η is defined in (53). The residual bound follows from Appendix B.4: $\|B_t\widetilde{w}_t - \widetilde{w}_{t-1}\|_{\mathrm{sp}} \le e_t + e_{t-1}$, whose sum is $2S_0 - e_0$ because $e_T = 0$. The active-request bound instead sums the exact Bellman gaps in (9). Empty requests leave the state unchanged and contribute zero, while the terminal error is zero. Thus (56) is the value-distortion bound for exact landmark values, with the known empty rounds removed.

Per-date checks and same-state action comparison. Bounds are checked for each date and selection seed before aggregation; the reported entries average each date's bound divided by its OPT. Oracle Full equality and zero residual and distortion at $m = n$ are checked as well.

For $K = 5$, an additional diagnostic compares predicted Full and Distortion 20% actions at each state actually visited by predicted Full. If a_t and b_t are their two actions from that same state s_{t-1}^F, compute

$$\Delta Q_t = d(s_{t-1}^F, b_t) + w_t(b_t) - d(s_{t-1}^F, a_t) - w_t(a_t).$$

Negative values favor the compressed action under exact continuation. The reported statistic averages $\sum_t \Delta Q_t/\mathrm{OPT}$ across dates and seeds. Because states are drawn from Full's trajectory, this sum is not the difference in the two policies' realized trajectory costs.

F.8 Synthetic pilot data and split independence

The metric is $M = \{x_j = j/8 : j = 0, \dots, 8\}$ with $d(x_i, x_j) = |x_i - x_j|$, diameter $D = 1$, and $s_0 = x_4 = 0.5$. Every episode has $T = 12$ tasks. For each of the seeds $7, 19, 41$, a fresh NumPy `default_rng(seed)` generator creates the training, validation, and test splits sequentially, with sizes 32, 64, and 256. Each scenario restarts the generator with the same seed values. No external dataset or preprocessing is involved.

The service costs are

$$c_{it}(x_j) = A_{it}(x_j - \mu_{it})^2 + U_{itj}, \qquad \mu_{it} = \mathrm{clip}_{[0,1]}(\bar{\mu}_t + \xi_i + \zeta_{it}), \tag{57}$$

where $\xi_i \sim \mathcal{N}(0, 0.045^2)$, $\zeta_{it} \sim \mathcal{N}(0, 0.035^2)$, and $U_{itj} \sim \mathrm{Unif}[0, 0.025]$. All underlying draws are independent, except that the same episode shift is deliberately shared across its tasks. The preferred center is clipped to $[0, 1]$; the resulting costs are already nonnegative and receive no further normalization. Generator phase $t = 0, \dots, 11$ corresponds to paper task $t + 1$.

Table 3: Complete pilot configuration. Gaussian scales are standard deviations. Both scenarios use the same split sizes and landmark budget.

Quantity	Localized	Switching
States / horizon / pair budget	$9 \,/\, 12 \,/\, 2$	$9 \,/\, 12 \,/\, 2$
Train / validation / test per seed	$32 \,/\, 64 \,/\, 256$	$32 \,/\, 64 \,/\, 256$
Preferred-center template $\bar{\mu}_t$	$0.68 + 0.11 \sin(2\pi t/6)$	$0.50 + 0.30 \sin(2\pi t/6)$
Episode / task Gaussian scales	$0.045 \,/\, 0.035$	$0.045 \,/\, 0.035$
Amplitude distribution	$\mathrm{Unif}[1.5, 2.5]$	$\mathrm{Unif}[4, 7]$
Per-state additive noise	$\mathrm{Unif}[0, 0.025]$	$\mathrm{Unif}[0, 0.025]$
Enumerated pair candidates	$\binom{9}{2} = 36$	$\binom{9}{2} = 36$

F.9 Label computation, fitting, and tie rules

The label routine `bellman` evaluates (1) in double precision. It stores a $(T+1) \times n$ array with a zero terminal row. The fitting routine anchors the first landmark in each sorted set and solves (43) using SciPy's `linprog` with `method='highs'`. The implementation uses sparse constraint matrices, the solver's default options, and checks solver success; there is no tuning on validation or test data. The returned table has shape $(T+1) \times m$, with only rows $1, \ldots, T-1$ optimized and unused time-zero and terminal rows set to zero. All-zero tables are returned directly for a singleton.

The pilot constructs pairs in lexicographic order. NumPy's first-index minimum rule resolves exact ties in pair selection and online actions; states are ordered by index. If the LP has multiple minimizers, the solver-returned optimum is used without a secondary objective. The locked environment records the implementation that produced the archived outputs; different solver versions can choose different optimal tables.

Selection methods. Each of the 36 pairs is fitted using only the training split. The distortion method adds the empirical κ average to the LP's empirical span loss and chooses a minimizing pair, implementing Theorem 14 up to numerical optimization error. The validation method freezes those fitted tables and chooses the pair with the lowest mean validation excess, as in Theorem 15. The geometric baseline minimizes, in order, covering radius, total distance of all states to the pair, and the pair's lexicographic index; it always selects $\{x_1, x_6\} = \{1/8, 3/4\}$. The random baseline is the arithmetic average of the test means of all 36 fitted pairs. It represents the exact expected performance of a uniformly drawn pair and does not select using test outcomes.

The singleton baseline compares all nine anchored zero-table policies on validation episodes. The full-state baseline fits $L = M$ using the same training split and LP, with no additional input information. All selection is completed before test performance is inspected.

Information available to each policy. A fitted rollout uses only its fixed table, the metric, its current state, and the current service-cost vector. Exact test values are used offline for OPT and the certificates below. The code's explicit `v=None` mode instead supplies exact landmark values at each time; only the rows labeled oracle use this mode. Every rollout uses $\widehat{w}_T = 0$ on all states.

F.10 Metrics, certificates, and aggregation

For every test episode, the primary metric is total episode excess $e(X) = \mathrm{ALG}(X) - w_0^X(s_0)$, not excess divided by T or by OPT. For seed s, let $\bar{e}_s = 256^{-1} \sum_{i=1}^{256} e(X_{si})$. The entries in Table 14 are

$$\bar{e} = \frac{1}{3} \sum_s \bar{e}_s, \qquad \mathrm{sd}_{\mathrm{seed}} = \sqrt{\frac{1}{3-1} \sum_s (\bar{e}_s - \bar{e})^2}. \tag{58}$$

Thus the uncertainty shown is the sample standard deviation of three seed means, not a confidence interval or the standard deviation of the 768 pooled episode costs. No hypothesis test is reported.

The recorded offline certificates are

$$C_\kappa(X) = \sum_{t=1}^{\tau} [\kappa_t(L; X) + 2\delta_t],$$
$$C_{\mathrm{loc}}(X) = \sum_{t=1}^{\tau} \min\{2D,\, 2\rho_t^\star(s_{t-1}; L) + 2\delta_t\}. \tag{59}$$

Their validity follows from Theorems 9 and 10 and the diameter bound. The implementation identifies numerical Bellman minimizers with `isclose` using absolute and relative tolerances 10^{-10}. Online actions themselves use the unmodified first-index argmin. Pilot certificate inequalities are checked with additive tolerance 10^{-8}; these checks accommodate floating-point arithmetic rather than changing the mathematical definition of a successor.

F.11 Software and reproduction

The Citi Bike runs used Python 3.11.6, NumPy 1.26.4, pandas 2.3.3, PyTorch 2.5.1 and CUDA 11.8. Mean-predictor runs used two A100 80GB GPUs, neural training used one, and additional resolution/oracle runs used CPUs with four threads per process. DP and evaluation use float64; neural training and greedy landmark scoring use float32. Synthetic DP and LP runs used Linux x86_64 CPUs with Python 3.13.5, NumPy 2.3.5, SciPy 1.17.0 and Matplotlib 3.10.8.

The accompanying code package's `README.md` and `docs/REPRODUCING.md` describe setup, data preparation and execution. Run `bash scripts/reproduce_synthetic.sh` to rerun both synthetic pilots and implementation checks, and compare with `reference_results/synthetic/`. Raw Citi Bike data require a separate download; trained weights are not included. In the paper repository, `scripts/build_a100_assets.py` regenerates tables, plots and supplementary CSV files from the frozen report tables in `data/`, without retraining or recomputing date-level statistics. `supplementary/README.md` indexes the detailed numerical results and implementation records.

G Additional Experimental Results

We report budget and resolution effects, compression across predictors, exact-value diagnostics, sensitivity checks, and synthetic fixed-table learning under the protocols in Appendix F. Complete numerical sweeps and implementation records accompany the paper in the supplementary files described in Appendix F.11.

G.1 Prediction budgets and resolution

Figure 3 combines the predicted and exact-value budget sweeps across all eight settings. Small budgets can fail: for $K = 4$ at 15 minutes, retaining $10/210$ predicted values gives Distortion cost ratio 1.8126, above WFA's 1.7067. At $63/210$, Distortion and Geometric reach 1.1527 and 1.1528, compared with Full's 1.1524. The curves also retain nonmonotone behavior across budgets.

At approximately 20%, the one-minute Distortion–Full mean differences for $K = 3, 4, 5, 6$ are $-0.001747, -0.000142, -0.000293, 0$. Table 4 gives their paired intervals and cross-resolution contrasts; each resolution is normalized by its own OPT. For $K = 5$ at one minute, the block interval $[-0.000662, -0.000011]$ excludes zero, whereas ordinary date resampling gives $[-0.000680, +0.000009]$. The apparent significance of this tiny difference is sensitive to the resampling scheme. For $K = 6$, the recorded zero gap and $[0, 0]$ interval apply to the evaluated dates and seeds. Table 5 reports comparisons against Random and Geometric for the five settings with archived paired selector comparisons.

Reconstruction and Bellman-residual diagnostics. Table 6 compares Full and Distortion reconstruction span and Bellman residual. Complete four-method diagnostics and marginal intervals are retained in `supplementary/results/a100_errors.csv`. For $K = 5$, Distortion reduces ALG/OPT from 1.3810 to 1.2737 even though normalized residual rises from 15.5573 to 39.5873. These diagnostics need not rank policies by online cost. Reconstruction span is measured against predicted Full values; exact-value distortion and its cost certificates are evaluated in Appendix G.3.

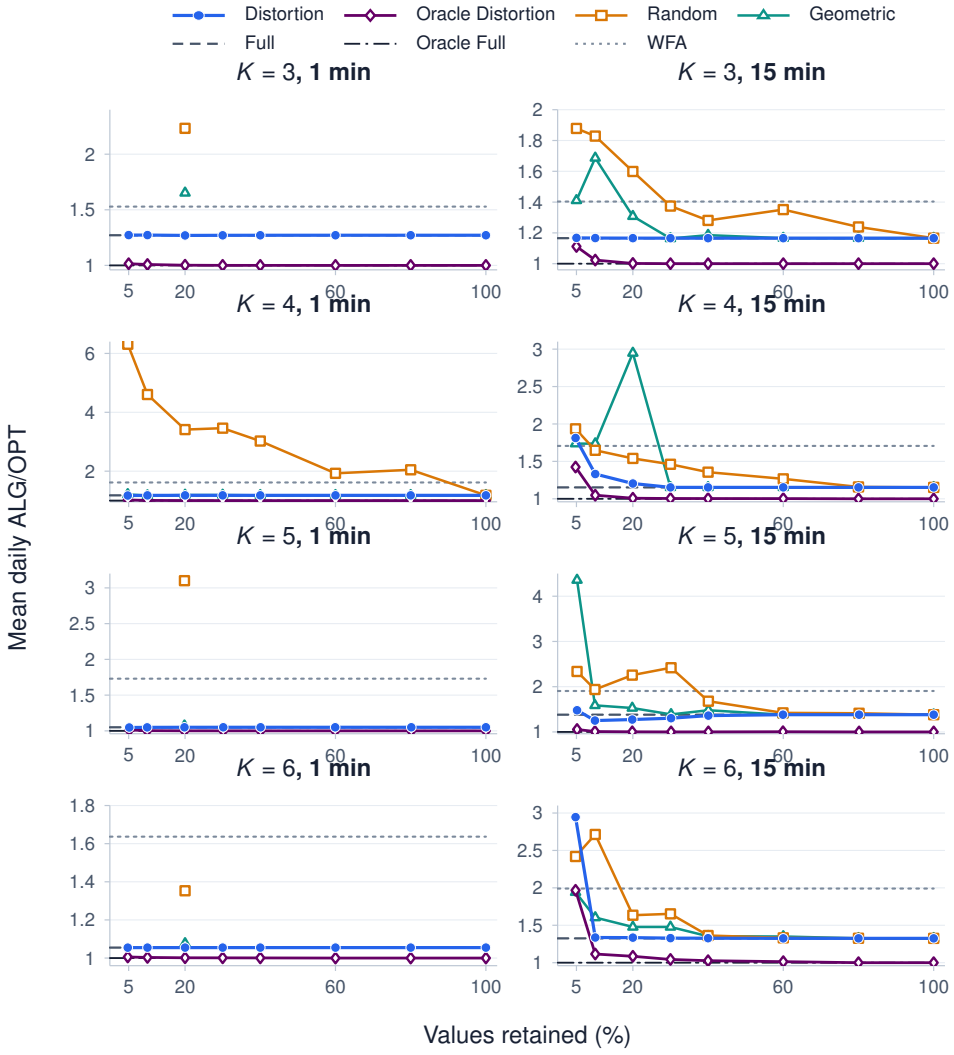

Figure 3: Predicted and exact-value budget–cost curves across all eight settings. Distortion uses the same frozen landmarks for predicted and oracle inputs; Oracle Full equals OPT. Random and Geometric have complete sweeps in five settings and only the reported 20% points in the other three one-minute settings. Lines connect supplied observations; axis ranges differ. Exact-value results are discussed in Appendix G.3.

Table 4: Distortion minus Full with approximately 20% of predicted values retained, and its paired cross-resolution contrast. Each resolution uses its own daily OPT. All intervals here use circular seven-day blocks, 2,000 replicates, seed 20260922.

K	Resolution / contrast	Δ or contrast	95% block CI
3	1 min	-0.001747	[-0.003456, -0.000187]
3	15 min	+0.000169	[+0.000000, +0.000506]
3	1 min minus 15 min	-0.001916	[-0.003676, -0.000297]
4	1 min	-0.000142	[-0.000862, +0.000551]
4	15 min	+0.0518	[+0.0306, +0.0728]
4	1 min minus 15 min	-0.0520	[-0.0729, -0.0306]
5	1 min	-0.000293	[-0.000662, -0.000011]
5	15 min	-0.1074	[-0.1733, -0.0401]
5	1 min minus 15 min	+0.1071	[+0.0401, +0.1730]
6	1 min	+0.000000	[+0.000000, +0.000000]
6	15 min	+0.0097	[-0.0075, +0.0265]
6	1 min minus 15 min	-0.0097	[-0.0265, +0.0075]

Table 5: Paired selector differences at approximately 20%. Negative differences favor Distortion. Intervals use 2,000 seven-day block replicates (seed 20260919) and are unadjusted for multiple comparisons. Distortion–Full intervals are in Table 4.

Setting	Comparison	Difference	95% block CI
$K = 3$, 15 min	Distortion - Random	-0.43294	[-0.4845, -0.3891]
$K = 3$, 15 min	Distortion - Geom.	-0.14221	[-0.2242, -0.0664]
$K = 4$, 15 min	Distortion - Random	-0.33533	[-0.3843, -0.2904]
$K = 4$, 15 min	Distortion - Geom.	-1.74355	[-1.8554, -1.6376]
$K = 5$, 15 min	Distortion - Random	-0.98201	[-1.0892, -0.8776]
$K = 5$, 15 min	Distortion - Geom.	-0.25517	[-0.3431, -0.1614]
$K = 6$, 15 min	Distortion - Random	-0.29947	[-0.3421, -0.2657]
$K = 6$, 15 min	Distortion - Geom.	-0.14265	[-0.1835, -0.1017]
$K = 4$, 1 min	Distortion - Random	-2.23429	[-2.3398, -2.1324]
$K = 4$, 1 min	Distortion - Geom.	-0.01089	[-0.0201, -0.0015]

Table 6: Mean-predictor diagnostics for Full and approximately 20% Distortion. Recon. is the per-slot reconstruction span relative to predicted Full values. The supplementary CSV retains all four methods and their ordinary date-bootstrap marginal intervals.

Setting	Method	ALG/OPT	η/OPT	Recon.
$K = 3$, 15 min	Full	1.1657	7.6815	0.0000
$K = 3$, 15 min	Distortion	1.1659	9.4318	0.8925
$K = 4$, 15 min	Full	1.1524	9.7615	0.0000
$K = 4$, 15 min	Distortion	1.2043	18.2262	1.4162
$K = 5$, 15 min	Full	1.3810	15.5573	0.0000
$K = 5$, 15 min	Distortion	1.2737	39.5873	1.6323
$K = 6$, 15 min	Full	1.3252	14.2701	0.0000
$K = 6$, 15 min	Distortion	1.3349	44.0859	1.9097
$K = 4$, 1 min	Full	1.1788	8.8766	0.0000
$K = 4$, 1 min	Distortion	1.1786	9.9711	0.1928

G.2 COMPRESSION ACROSS PREDICTORS

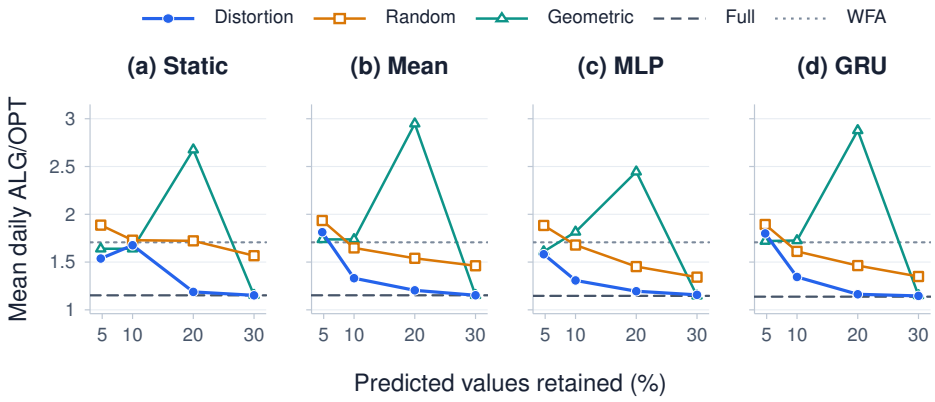

Figure 4: Compression across predictors for $K = 4$ at 15 minutes. Panels share a vertical scale and the styles in Figure 2. Points show all four tested compressed budgets; every method recovers Full at $m = n$, omitted here.

Tables 7 to 9 separate Full-predictor quality from landmark selection. The GRU has the lowest Full cost point estimate, but its paired intervals against the mean table, MLP and static potential all cross zero. At the 20% budget, its Distortion–Random and Distortion–Geometric intervals are wholly negative. Thus the evidence for landmark selection is clearer than the evidence for improving the predictor itself. All training and landmark seeds are averaged; none is selected on test cost.

Table 7: Full-predictor costs and value errors on the 365 evaluation dates. RMSE uses centered exact values; value-error span and Bellman residual are distinct diagnostics.

Predictor	Full	95% block CI	RMSE	Error span	η/OPT
Static	1.1520	[1.1202, 1.1863]	1.0403	4.3440	12.4987
Mean	1.1524	[1.1214, 1.1865]	0.7726	2.9219	9.7615
MLP	1.1464	[1.1207, 1.1736]	0.7856	3.2413	15.5222
GRU	1.1378	[1.1139, 1.1632]	0.7649	3.0593	14.5484

Table 8: Temporal-family compression retaining $m = 42$ of 210 predicted values. Neural results average the crossed training and landmark seeds within each date. Bold marks the lowest ALG/OPT within each predictor, without implying statistical significance.

Predictor	Method	ALG/OPT	95% block CI	Ret.	Recon.
Static	Distortion	**1.1877**	[1.1576, 1.2201]	93.57%	1.5840
Static	Random	1.7223	[1.6565, 1.7925]	-2.81%	3.9430
Static	Geom.	2.6758	[2.5404, 2.8136]	-174.69%	3.2623
Mean	Distortion	**1.2043**	[1.1826, 1.2259]	90.65%	1.4162
Mean	Random	1.5396	[1.4922, 1.5905]	30.15%	3.9819
Mean	Geom.	2.9478	[2.8428, 3.0563]	-223.90%	3.0379
MLP	Distortion	**1.1950**	[1.1752, 1.2156]	91.33%	1.5246
MLP	Random	1.4528	[1.4087, 1.5016]	45.31%	3.9612
MLP	Geom.	2.4448	[2.3552, 2.5323]	-131.72%	2.9447
GRU	Distortion	**1.1630**	[1.1386, 1.1884]	95.57%	1.5136
GRU	Random	1.4633	[1.4208, 1.5119]	42.78%	3.9983
GRU	Geom.	2.8774	[2.7675, 2.9905]	-205.76%	2.9441

Static potential achieves a mean daily ALG/OPT close to the mean table's in this setting, despite removing its time variation and having larger centered RMSE and value-error span. The GRU reduces centered MSE by 1.97% relative to the mean table, but its value-error span and Bellman residual increase. Training minimizes MSE, so improvement in all error measures is neither enforced nor observed. The mean-predictor rows are shared with the preceding comparisons.

Table 9: Paired temporal-family comparisons. Intervals use seven-day block resampling and condition on the fitted data and seeds.

Comparison	Difference	95% block CI
GRU Full - Mean Full	-0.01463	[-0.0446, +0.0127]
GRU Full - MLP Full	-0.00860	[-0.0210, +0.0034]
GRU Full - Static Full	-0.01417	[-0.0442, +0.0134]
MLP Full - Mean Full	-0.00604	[-0.0328, +0.0182]
Static Full - Mean Full	-0.00046	[-0.0019, +0.0009]
Static: Distortion 20% - Full	+0.03569	[+0.0281, +0.0436]
Mean: Distortion 20% - Full	+0.05182	[+0.0302, +0.0730]
MLP: Distortion 20% - Full	+0.04859	[+0.0273, +0.0695]
GRU: Distortion 20% - Full	+0.02522	[+0.0100, +0.0420]
GRU: Distortion 20% - Random 20%	-0.30032	[-0.3495, -0.2571]
GRU: Distortion 20% - Geom. 20%	-1.71436	[-1.8271, -1.6035]

G.3 Exact future-value compression and theoretical certificates

Table 10 holds the landmark sets fixed while replacing their predicted dual values by exact evaluation-day values. Oracle Full equals OPT; using exact values only at landmarks can still incur excess. For $K = 6$ at 15 minutes, Oracle Distortion at approximately 20% has cost ratio 1.0854, with 95% block interval $[1.0738, 1.0970]$. The corresponding one-minute ratio is 1.0014. For $K = 4$, the per-slot exact distortion is 0.1423 at one minute versus 1.3849 at 15 minutes, and oracle ratios are 1.0015 and 1.0077. The resolution contrast thus persists under exact dual-value inputs, although this comparison does not separate the effects of demand, horizon, or the fitted landmark sets.

Table 10: Frozen-landmark exact-value diagnostic at approximately 20%. Oracle Full equals OPT in every setting. Intervals use seven-day blocks. The last column replaces inputs at fixed landmarks; it is not an independent causal component of total loss.

Setting	Oracle Distortion [95% CI]	Predicted minus oracle [95% CI]
$K = 3$, 1 min	1.0017 [1.0013, 1.0022]	+0.2678 [+0.2524, +0.2844]
$K = 3$, 15 min	1.0016 [1.0007, 1.0026]	+0.1643 [+0.1404, +0.1889]
$K = 4$, 1 min	1.0015 [1.0009, 1.0022]	+0.1771 [+0.1627, +0.1939]
$K = 4$, 15 min	1.0077 [1.0050, 1.0108]	+0.1966 [+0.1760, +0.2191]
$K = 5$, 1 min	1.0005 [1.0001, 1.0011]	+0.0494 [+0.0420, +0.0572]
$K = 5$, 15 min	1.0030 [1.0014, 1.0050]	+0.2707 [+0.2307, +0.3187]
$K = 6$, 1 min	1.0014 [1.0001, 1.0033]	+0.0534 [+0.0441, +0.0634]
$K = 6$, 15 min	1.0854 [1.0738, 1.0970]	+0.2495 [+0.2077, +0.2897]

The $K = 5$ same-state diagnostic in Table 11 evaluates action changes on the states visited by predicted Full. At 15 minutes, the net exact-continuation difference is -0.0832, with interval $[-0.1311, -0.0360]$, supporting the explanation that compression can alter decisions induced by an imperfect Full predictor beneficially. Improved actions are not more frequent than worsened actions; their magnitudes also matter. At one minute, actions differ only about 0.01 times per day and the net difference interval crosses zero. This diagnostic is trajectory-conditioned, not a causal or additive decomposition of total cost.

Table 11: $K = 5$ actions compared at the same states visited by predicted Full. Counts are daily averages; the final column averages the daily normalized sum of exact-continuation action differences. This is not the cost difference between the two policies' actual trajectories.

Minutes	Changed	Improved	Worsened	$\sum \Delta Q/\mathrm{OPT}$ [95% CI]
1 min	0.010	0.004	0.002	-0.000147 [-0.000493, +0.000110]
15 min	0.992	0.336	0.348	-0.0832 [-0.1311, -0.0360]

Appendix F.7 defines the distortion indexing and recorded certificates. The bounds in Table 12 hold separately for every recorded seed/date, but are generally much larger than realized excess. For

Table 12: Exact-value distortion and certificates for Distortion at approximately 20%. Normalize within each date before averaging. The final three columns are cost-ratio upper bounds, not attained costs; Appendix F.7 defines $S_0, B_{\mathrm{adj}}, B_{\mathrm{act}}$.

Setting	$\bar{e}$	S_0/OPT	R_{or}	$1+\eta/\mathrm{OPT}$	$1+B_{\mathrm{adj}}/\mathrm{OPT}$	$1+B_{\mathrm{act}}/\mathrm{OPT}$
$K=3$, 1 min	0.1848	1.450	1.0017	2.043	3.900	2.403
$K=3$, 15 min	0.7715	8.138	1.0016	5.207	17.238	9.099
$K=4$, 1 min	0.1423	3.506	1.0015	3.230	8.010	4.408
$K=4$, 15 min	1.3849	19.542	1.0077	16.962	40.004	20.458
$K=5$, 1 min	0.1725	9.965	1.0005	6.581	20.929	10.815
$K=5$, 15 min	1.6489	41.135	1.0030	35.985	82.980	41.838
$K=6$, 1 min	0.2214	20.694	1.0014	10.512	42.384	21.406
$K=6$, 15 min	1.9219	50.474	1.0854	40.255	101.540	51.058

example, $K=4$ at one minute has actual oracle ratio 1.0015, versus residual-based bound 3.230 and active-round distortion bound 4.408. These checks connect the implementation to the upper bounds, without establishing empirical tightness.

Figure 3 retains all eight Distortion budgets in all eight settings. Oracle compression cost need not decrease monotonically: for $K=5$ at 15 minutes, its ratio is 1.0007 at $76/252$ retained values and 1.0044 at $151/252$, before reaching OPT at Full.

G.4 Training-data and date sensitivity

For $K=4$, 15-minute requests and $m=42$, varying fitting days along one nested sampling path gives nonmonotone Distortion costs; the 731-day result is worse than several smaller samples. Calibration still uses 639 days in every condition, so this is a sensitivity check rather than empirical sample-complexity scaling. The protocol is in Appendix F.5, and the complete results are retained in `supplementary/results/a100_data_sensitivity.csv`.

Table 13 combines the eight mean-predictor settings. Excluding the first ten evaluation dates leaves the observed costs close to the full-year values.

Table 13: Mean daily ALG/OPT at approximately 20% after excluding the first ten 2025 dates. All eight settings use the same fitted models and landmarks as their full-year evaluations.

Setting	365-day Distortion	355-day Distortion
$K=3$, 1 min	1.2695	1.2707
$K=3$, 15 min	1.1659	1.1688
$K=4$, 1 min	1.1786	1.1797
$K=4$, 15 min	1.2043	1.2054
$K=5$, 1 min	1.0499	1.0494
$K=5$, 15 min	1.2737	1.2743
$K=6$, 1 min	1.0548	1.0539
$K=6$, 15 min	1.3349	1.3359

G.5 Synthetic fixed-table learning and cost certificates

Table 14 reports the two fixed-table pilots under the data and fitting protocol in Appendices F.8 and F.9.

The localized singleton matches the learned pair and fitted Full table; this workload does not establish a benefit from additional landmarks. Switching demand exhibits a nontrivial tradeoff between pair and Full prediction.

Table 15 gives the two offline certificates in (59) for comparison with these observed costs. The bounds can be substantially larger than realized excess. For localized learned-validation pairs, mean excess is 0.04810, whereas the mean distortion and local certificates are 1.51201 and 0.49766, respectively. The local certificate uses only optimal successors from the actual online states, which

Table 14: Test total episode excess $\mathrm{ALG} - \mathrm{OPT}$: mean $\pm$ sample standard deviation of the three seed-level means (not confidence intervals). Oracle full-state excess is below 10^{-15} in magnitude and displayed as zero. Bold marks the lowest non-oracle mean in each column, including ties, without implying statistical significance.

Method	Localized	Switching
Uniform random pairs (exact average)	0.40154 ± 0.01142	0.43569 ± 0.00668
Geometric pair	0.17838 ± 0.01720	0.39766 ± 0.01932
Learned distortion pair	**0.04810** ± 0.00666	0.13927 ± 0.01661
Learned validation pair	**0.04810** ± 0.00666	0.13927 ± 0.01661
Learned singleton	**0.04810** ± 0.00666	0.41436 ± 0.00850
Full-state fixed table	**0.04810** ± 0.00666	**0.04031** ± 0.00882
Oracle geometric pair	0.17838 ± 0.01720	0.43890 ± 0.02160
Oracle learned-validation pair	0.04810 ± 0.00666	0.15341 ± 0.01725
Oracle full state	0	0

can explain a sharper bound than the all-state distortion in this setting. Neither bound universally dominates the other.

Table 15: Mean offline certificates over the three seed-level test means. Observed excess and seed standard deviations are in Table 14; C_κ and C_{loc} are defined in (59).

	Localized		Switching	
Method	C_κ	C_{loc}	C_κ	C_{loc}
Geometric pair	3.05480	1.72344	3.94197	4.21647
Learned distortion pair	1.15931	0.60406	2.13124	1.79679
Learned validation pair	1.51201	0.49766	2.13124	1.79679
Learned singleton	1.30334	0.28874	4.05037	3.42253
Full-state fixed table	0.94929	0.94929	0.98307	0.98307
Oracle geometric pair	2.16176	0.83040	3.00155	3.27604
Oracle learned-validation pair	1.30308	0.28874	1.67918	1.34505
Oracle full state	0	0	0	0

For a full-state fitted table, exact reconstruction distortion is zero and every optimal successor is a landmark. Its two recorded certificates therefore agree and measure prediction error. For the oracle full-state policy both vanish, and the recorded excess is zero up to roundoff. In contrast, oracle compressed values can yield larger excess than a fitted compressed table: switching mean excess is 0.15341 for the oracle learned-validation pair and 0.13927 for its fitted counterpart. Exact dual values do not remove the envelope's reconstruction bias, and the upper bounds do not assert monotonicity of realized cost in landmark prediction error.

G.6 Implementation checks and relation to learning guarantees

Archived checks cover DP against exhaustive enumeration, causal predictor inputs, Lipschitz outputs, terminal zero, landmark interpolation, Full recovery, LP loss, and per-episode cost certificates. The theoretical constructions were also checked numerically. These implementation checks passed in the archived runs; detailed configurations and outcomes are retained with the supplementary records.

The synthetic pilots implement the fixed-table LP and joint selection directly. In Citi Bike, the practical Distortion selector omits the fitted-table error term, the mean predictor does not minimize span loss, and contextual neural predictors minimize centered MSE. Those runs therefore do not instantiate the joint-ERM PAC guarantee.