EXACT MINIMUM-EVENT EXPLANATIONS UNDER LOCALIZED OBSERVATION UNCERTAINTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Changepoint detectors, trend filters and perception pipelines routinely propose sparse event explanations of time series. We study how to verify such proposals exactly: given interval-valued observations of a scalar trajectory whose velocity jumps are nonnegative, capped and confined to labelled timing windows, can at most k events explain them? The budget query is NP-complete in general. We identify a shared-endpoint reset: across an interval with exactly observed endpoints and a timing window that opens at its start, the full set of feasible outgoing velocities is one interval, whatever the incoming set. The reset confines combinatorial branching to uncertainty blocks of maximum length β and gives an exact algorithm in $2^{\beta}\,\mathrm{poly}(n, \beta, L)$ time with rational witness recovery; assuming the Exponential Time Hypothesis, no $2^{o(\beta)}$ algorithm exists. A five-interval construction shows why exactness is needed: a complete-history first-moment flow relaxation with full-horizon fractional lookahead accepts two events although every compatible trajectory needs three. In the verified comparisons, every ACCEPT has a rational witness and every REJECT an exact certificate. On 24 retail-log windows fixed before inference, local inference settles 666 of 672 budget queries versus 540 for a bound-based baseline; on two taxi-log panels it matches enumeration and a certifying MILP cascade at about 28% less CPU than enumeration. At fixed block size two, a separate prospective synthetic panel with n up to 130 yields 14 of 16 certified budget decisions, retaining two call-limited UNKNOWNs. Exact certificates also expose over-counting and model-inconsistent proposals from ℓ_1 relaxations and trend filtering. We report null comparisons and a downstream pilot without local-inference gains.

1 INTRODUCTION

Learned and statistical models routinely propose sparse event explanations of time series. A changepoint detector reports breakpoints, a trend filter reports slope changes, and a tracker that outputs positions with error bars implies a number of impacts. Before such an explanation is used, it is natural to ask a *budget query*: can at most k events explain everything that was observed? An exact answer has two sides. A compatible trajectory with at most k events certifies that the budget suffices; showing that it is too small requires excluding every such trajectory. Convex relaxations can accept explanations that no trajectory realizes, and floating-point solvers can report infeasibility without proof. We ask when both sides can be certified exactly and efficiently.

Figure 1 shows a small instance. Four recorded retail invoices fall into three ten-minute bins, but after two checkpoints are coarsened, the integrated count admits an explanation with two occupied bins. Our local dynamic program certifies that two events suffice, which a bound-based front end leaves unresolved. The answer concerns compatible explanations; it does not estimate the latent invoice sequence.

We study a scalar model with known acceleration between nonnegative, capped velocity jumps, observed through rational position intervals, with at most one jump per labelled sampling interval inside a prescribed timing window. After detrending, trajectories are piecewise linear with nonnegative slope increases, so this is a constrained form of fitting a piecewise-linear signal with few breakpoints (Kim et al., 2009; Fearnhead et al., 2019; Zhao et al., 2022). The same model describes integrated arrival counts exactly. Budget queries are NP-complete (Proposition 4), and the difficulty

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.

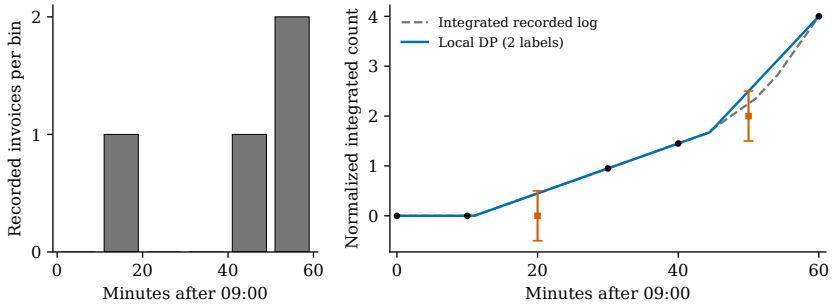

Figure 1: A budget query derived from a recorded log (UCI Online Retail, January 5, 2011, 09:00–10:00). Left: four invoices occupy three ten-minute bins. Right: the integrated count, normalized by the largest bin count, with exact checkpoints (dots), two quantization cells of half-width $1/2$ (bars) and a checked two-event Local DP explanation. The exact minimum is two occupied labels.

is continuous: a forward dynamic program must carry, for each budget, the set of velocities reachable by a compatible prefix, which is a union of intervals contributed by different support patterns.

Our main positive result is a *reset* that removes this branching at exactly observed intervals. If an interval has exact endpoint positions and its timing window opens at its left endpoint, every feasible transition across it reaches the same smallest outgoing velocity, the chord slope, so the union of transition images is one interval even when the incoming set is disconnected. Branching is therefore confined to maximal runs of intervals that touch an uncertain observation, and an exact algorithm runs in time exponential only in the longest run β. A complementary negative result explains why exact continuous compatibility is needed. In a layered history-flow relaxation that passes position and velocity moments, a node with one complete event history can redistribute velocity among outgoing copies that no compatible prefix with that history can reach, even with full-horizon fractional lookahead. The missing ingredient is the association between each continuous state and a feasible prefix, not more discrete memory.

Contributions. (i) An exact $2^{\beta}\,\mathrm{poly}(n, \beta, L)$ algorithm with rational witness recovery under an explicit reset condition (Theorem 6), a joint-block extension for local timing restrictions, NP-completeness without structure, and an ETH lower bound showing that the 2^{β} factor cannot be improved to $2^{o(\beta)}$ (Corollary 7). (ii) An explicit rational family on which a complete-history first-moment flow with full-horizon fractional lookahead accepts budget two while the physical minimum is three (Theorem 2). (iii) Certified experiments on designed and log-derived queries: Local DP settles budgets left open by bounds, matches enumeration and a certifying MILP cascade at lower cost than enumeration, and settles 14 of 16 fresh structured queries with n up to 130 at fixed block size two; exact certificates also audit event counts proposed by ℓ_1 relaxations and trend filtering. Null comparisons and a downstream pilot without Local DP gains delimit these results.

2 Related work

Segmentation and sparse slope changes. Exact dynamic programs compute penalized change-point segmentations with pruning (Killick et al., 2012; Maidstone et al., 2017), including constraints between segments (Hocking et al., 2022). CPOP fits continuous piecewise-linear means with an L_0 penalty on slope changes by conditioning on a continuous endpoint value (Fearnhead et al., 2019), and ℓ_1 trend filtering relaxes that penalty convexly (Kim et al., 2009; Tibshirani, 2014). Bounded-error identification seeks the fewest affine submodels or switches consistent with interval-bounded errors, for example by ℓ_1 sparsification (Bemporad et al., 2005; Ozay et al., 2012). Geometric algorithms fit lines and polygonal functions within vertical tolerances (O'Rourke, 1981; Hakimi & Schmeichel, 1991), and recent work minimizes segment counts under error bounds by linear-time algorithms or mixed-integer programs (Zhao et al., 2022; Ploussard, 2024; Warwicker & Rebennack, 2025; Magnani & Boyd, 2009). These methods optimize a fit criterion or relax the count. We answer a hard-feasibility budget query under capped nonnegative jumps and per-label timing windows, with certificates for both answers. Continuous-state dynamic programming is established; our point is an exact collapse of its feasible-set union and the parameter it yields.

Formulations and certification. For a fixed support the feasible set is polyhedral, so the problem is a union of polyhedra (Vielma, 2018); our exact-prefix comparator uses this disaggregation. Automaton-based flow formulations are exact for regular languages and extend to continuous control languages (Fiorini & Pashkovich, 2015; Buchheim & Merkert, 2025); our negative result concerns a weaker first-moment architecture and does not contradict them. A difference-of-convex MILP for general-dimensional PLA preserves sampled values but not our jump labels, windows and cap (Ploussard et al., 2026). We pair floating-point proposals with exact rational verification (Applegate et al., 2007; Gleixner et al., 2016). Certifying MILP infeasibility would require verified branch-and-bound (Cheung et al., 2017), so numerical MILP exclusions are reported separately. Appendix F gives an extended comparison.

3 Model and exact explanations

There are $n \geq 1$ labelled intervals of duration $D > 0$ and sampled positions $p_0, \ldots, p_n$, from which known acceleration has been subtracted. Interval i carries a jump $0 \leq j_i \leq J$ in velocity at delay $\tau_i \in [l_i, u_i] \subseteq [0, D]$ from its start. With w_i the velocity entering interval i and $r_i = (D - \tau_i)j_i$, compatibility is equivalent to

$$p_{i+1} - p_i = Dw_i + r_i, \qquad w_{i+1} - w_i = j_i, \tag{1}$$

$$0 \leq j_i \leq J, \qquad (D - u_i)j_i \leq r_i \leq (D - l_i)j_i, \tag{2}$$

$$a_i \leq p_i \leq b_i. \tag{3}$$

All data are rational, initial and final velocities are free, and L denotes the total encoding length of the data. For $j_i > 0$ the delay is $\tau_i = D - r_i/j_i$; for $j_i = 0$, (2) forces $r_i = 0$. We call a solution of (1)–(3) a *physical trajectory*, to distinguish it from solutions of relaxations.

Definition 1 (Minimum compatible explanation)**.** The support of a physical trajectory is $\{i : j_i > 0\}$, and OPT is its minimum cardinality ($+\infty$ if the observations are infeasible). The budget query asks whether $\mathrm{OPT} \leq k$ for an integer $k \geq 0$; algorithms need only $\bar{k} = \min\{k, n\}$.

Support counts *labelled* events: two events at adjacent window endpoints of intervals i and $i + 1$ count twice. For an allowed mask S, setting $j_i = r_i = 0$ outside S gives a rational LP that is feasible exactly when a physical trajectory supported within S exists; zero jumps remain allowed inside S. A physical witness with at most k positive labels certifies $\mathrm{OPT} \leq k$, and a sound exclusion of all supports of size at most k certifies $\mathrm{OPT} > k$. The support of the trajectory that generated an input is only an upper bound, and an unfinished computation certifies nothing. Infeasibility at every budget rejects the model itself.

4 Complete history does not preserve continuous correlation

A natural way to strengthen convex relaxations of event inference is to track discrete event histories. A layered graph has nodes (i, c, H), where i is the sample index, c the number of designated event labels, and H their complete ordered history. Inactive arcs preserve H and force $j_i = r_i = 0$; active arcs append i and increment c, with $c \leq k$. Each arc carries a nonnegative flow and mass-scaled position, velocity, jump and moment variables that satisfy the perspective form of (1)–(3). Unit flow travels from source to sink, and flow and the first position and velocity moments are conserved at internal nodes.

Each incoming-head and outgoing-tail copy is given its own lookahead LP, starting from that copy. The lookahead includes all remaining observations and dynamics but replaces event cardinality by continuous activations $0 \leq z_t \leq 1$ with $j_t \leq Jz_t$ and $\sum_t z_t \leq k - c$, in homogeneous form for mass-scaled copies. We denote this complete-history, full-horizon, first-moment relaxation by $\mathcal{F}_k$; Appendix A states its constraints. Conservation relates averages. It does not require each outgoing state to be reachable through its recorded prefix.

Theorem 2 (A complete-history false explanation)**.** *Fix rational $0 < \lambda < 1$, $D = J = 1$, and midpoint windows $l_i = u_i = 1/2$. For the five-interval observations*

$$p_0 = \frac{19}{8} + \frac{\lambda}{2}, \quad p_1 \in \left[\frac{13}{8}, \frac{15}{8}\right], \quad p_2 = \frac{7}{8}, \quad p_3 = \frac{1}{4}, \quad p_4 = p_5 = 0, \tag{4}$$

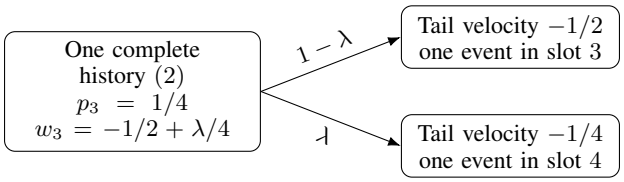

Figure 2: The split in Theorem 2. It preserves the mean velocity, yet for $0 < \lambda < 1$ neither outgoing velocity is reachable by a physical prefix with history (2). No two different histories are merged.

the physical minimum is $\mathrm{OPT} = 3$*, whereas* $\mathcal{F}_2$ *is feasible. At* $\lambda = 0$ *and* $\lambda = 1$ *the physical minimum is two.*

Proof sketch. Reverse time. The first four exact positions force the prefix jumps $(t, 1/2-t, 1/4+t)$ for some $0 \le t \le 1/2$. An interior t already uses three events. At either endpoint, two prefix events exhaust a budget of two, which forces a zero suffix and a terminal position of $19/8$ or $23/8$; neither equals the required value for $0 < \lambda < 1$. Taking $t = \lambda/2$ with a zero suffix gives a three-event witness.

In the original orientation, a common physical prefix reaches sample three with history $H = (2)$, position $1/4$ and velocity $-1/2 + \lambda/4$. The relaxation splits this state into masses $1 - \lambda$ and λ with velocities $-1/2$ and $-1/4$. The first tail uses one jump in slot three and the second one jump in slot four. Both reach the remaining positions, and their weighted velocity equals the incoming one. Fractional lookahead accepts the common prefix because its remaining activation sum is $1/2 \le 1$. Appendix B gives all witnesses. □

The counterexample (Figure 2) separates missing continuous correlation from insufficient discrete memory. It does not defeat full-horizon *exact-cardinality* lookahead, which would already solve the original problem.

Proposition 3 (Exact-prefix comparator)**.** *Attach a separate perspective copy of the exact fixed-history prefix polyhedron to every outgoing tail of every full-history node. The augmented flow is feasible if and only if a physical trajectory with at most* k *events exists.*

Proof. A physical trajectory embeds directly. Conversely, normalize a terminal arc with positive flow. Its attached prefix copy supplies a compatible complete prefix, and the terminal arc supplies the final interval. Gluing them at their common state gives a physical trajectory with at most the designated budget. □

The comparator can require exponentially many histories and prefix copies.

5 Exact inference from localized uncertainty

Proposition 4 (Hardness without structure)**.** *The budget query* $\mathrm{OPT} \le k$ *is NP-complete, even with* $D = J = 1$*, positive-width rational boxes at every sample, and positive-width timing windows contained in* $(1/4, 3/4)$.

Membership holds because a support can be checked with one masked LP; hardness is by reduction from exact-cardinality subset sum (Appendix D). The reduction is precision-dependent, so it does not give strong NP-hardness or hardness at a fixed noise level. In it every interval touches an uncertain sample; by Corollary 9 below, hardness indeed requires long blocks. Let

$$\mathcal{U} = \{i \in \{0, \dots, n\} : a_i < b_i\}, \qquad \mathcal{B} = \{i \in \{0, \dots, n-1\} : i \in \mathcal{U} \text{ or } i+1 \in \mathcal{U}\}. \tag{5}$$

An *uncertainty block* is a maximal run of consecutive intervals in $\mathcal{B}$, and β is the maximum block length ($\beta = 0$ if $\mathcal{B} = \varnothing$; Figure 3 in Appendix C gives an example). Intervals outside $\mathcal{B}$ are *regular*: both endpoint positions are exact. Our sufficient condition is

$$i \notin \mathcal{B} \quad \Longrightarrow \quad l_i = 0, \tag{6}$$

with arbitrary upper endpoints u_i and arbitrary windows inside blocks. The midpoint-window family of Theorem 2 violates it.

An exact reset. Consider a regular interval with chord slope $s = (p_{i+1}-p_i)/D$ and window $[0, u]$. Substituting $r = D(s - w)$ into (2) shows that an incoming velocity w is feasible iff $w \in [s - J, s]$, and that an allowed jump then maps it to the outgoing interval

$$[s, g_u(w)], \qquad g_u(w) = \min\{w+J,\ w+D(s-w)/(D-u)\}\ (u < D), \qquad g_D(w) = w+J. \tag{7}$$

Lemma 5 (Shared-endpoint reset)**.** *Let W be a closed finite union of incoming velocity intervals with $H = W \cap [s - J, s] \neq \varnothing$. Its allowed-jump image is exactly*

$$\bigcup_{w\in H}[s, g_u(w)] = [s, \max_{w\in H} g_u(w)], \tag{8}$$

and an inactive feasible branch contributes only the point s.

Proof. All intervals share the left endpoint s; the largest right endpoint is attained on compact H, and its interval contains the others. Inactivity gives $j = r = 0$, hence $w = w' = s$. □

This is an equality of physical images, not a convex hull: for example, with $D = J = 1$ and $s = u = 1/2$ the disconnected set $[-1/2, 0] \cup [1/4, 1/2]$ maps exactly onto $[1/2, 1]$. The zero lower endpoint is essential, since a jump at the interval's start reaches s from every feasible w. Writing T_i for the image map and $V_i(q)$ for the exact incoming set with support at most q, the regular recurrence is

$$V_{i+1}(q) = \big(\{s\} \text{ if } s \in V_i(q)\big)\ \cup\ T_i(V_i(q-1)), \qquad V_i(-1) = \varnothing. \tag{9}$$

The allowed branch charges one label but permits a zero jump, which never exceeds the charged budget.

Block transfer and complexity. At a regular cut let $I(q)$ be the exact reachable velocity interval with support at most q. For a block C and local mask $A \subseteq C$, let $\Phi_{C,A}(I)$ be the projection onto the outgoing velocity of the complete block LP with incoming velocity in I and labels outside A inactive; it is empty or a closed rational interval. Then

$$V_{\text{out}}(q) = \bigcup_{A\subseteq C,\ |A|\le q} \Phi_{C,A}\big(I(q - |A|)\big), \tag{10}$$

and the next regular interval collapses this union to one record.

Theorem 6 (Reset/block exactness and tractability)**.** *Under* (1)–(3) *and* (6)*, support at most k is decidable exactly with at most 2^β velocity-interval records per budget at block exits and one after each regular interval, using $O(n(\bar{k}+1)2^\beta)$ exact interval operations and $2^\beta \operatorname{poly}(n, \bar{k}, \beta, L)$ bit complexity. Minimum support is computable in $2^\beta \operatorname{poly}(n, \beta, L)$ time, and a rational physical witness is recovered by one masked LP.*

Proof sketch. Blocks and regular intervals are processed in order. Each block receives one record per predecessor budget and emits one interval per mask; Lemma 5 restores a single record at the next regular interval. Every physical trajectory is covered by the mask of its positive labels, and concatenation at exact boundary positions proves soundness. For bit complexity, every stored interval equals the projection of an original-coefficient prefix LP for one allowed mask, so endpoints are bounded by determinants of such systems rather than by a recurrence on computed endpoints (Appendix C). □

Corollary 7 (The exponential dependence on β is necessary)**.** *Assuming the Exponential Time Hypothesis, no algorithm decides* $\text{OPT} \le k$ *in $2^{o(\beta)} \operatorname{poly}(n, L)$ time, even on instances with $D = J = 1$ that satisfy* (6)*.*

The reduction behind Proposition 4 maps Subset Sum with d numbers to an instance with $\beta = n = 4d + 4$ that satisfies (6) vacuously, because no interval is regular; Subset Sum has no $2^{o(d)}$-time algorithm under ETH (Impagliazzo et al., 2001; Etscheid et al., 2015). Theorem 6 is therefore optimal up to the constant in the exponent (Appendix D).

Table 1: ACCEPT / REJECT / UNKNOWN counts over all queries. The retail baseline is Bounds + block-cap. Retail queries share windows and budgets, so they are correlated. Every decisive Local DP answer agrees with enumeration.

Method	D (6)	E (12)	Retail-I (224)	Retail-II (672)
Bounds	0 / 0 / 6	0 / 0 / 12	–	–
Bounds + block-cap	0 / 3 / 3	0 / 4 / 8	108 / 80 / 36	311 / 229 / 132
Local DP	**3 / 3 / 0**	**8 / 4 / 0**	142 / 81 / 1	431 / 235 / 6
Enumeration	2 / 2 / 2	4 / 2 / 6	142 / 82 / 0	433 / 239 / 0

Local timing restrictions and scope. Let $\mathcal{B}^+ = \mathcal{B} \cup \{i : l_i > 0\}$ and let β^+ be its longest run. Absorbing the extra intervals into enlarged blocks computes minimum support in $2^{\beta^+}\,\mathrm{poly}(n, \beta^+, L)$ time for arbitrary windows (Corollary 9); an exact-endpoint interval with $0 < l_i < D$ cannot act as a reset, because its least outgoing velocity depends on w. The parameter can stay small while the number of uncertain samples grows with n, but it reaches n when every observation is uncertain, consistent with Proposition 4. A separate separator algorithm handles arbitrary windows in $2^{(\bar{k}+1)\beta}\,\mathrm{poly}(n, \bar{k}, \beta, L)$ time (Appendix E); under (6) the reset removes the budget from the exponent.

6 Experiments

We ask three questions. (Q1) Does exact local inference settle budget queries that cheaper bounds leave open? (Q2) When certifying methods agree, what does a complete certified decision cost? (Q3) Where does the implementation stop, and why? Appendix G reports every study in full, with all denominators, figures and verification boundaries.

Protocol and methods. A method returns ACCEPT with a rational physical witness using at most k positive labels, REJECT with an exact original-model lower certificate (Farkas or rational structural) exceeding k, and UNKNOWN otherwise, including at resource limits. Floating-point solvers only propose; every decisive answer is recovered exactly and replayed without a solver. Except for a standalone MILP, methods share a front end, **Bounds**: the larger of an observation-derived packing lower bound and the LP-relaxation bound of the activation formulation, keeping a relaxation solution if it passes the exact check. **Block-cap** adds a structural bound from exact affine triples and the jump cap. The continuations are **Local DP** (Section 5), exhaustive **enumeration** of original-model supports, and an activation **MILP** with binary z_i, $j_i \le J z_i$ and $\sum_i z_i \le k$, whose positive answers are accepted only after an exact witness check. When every observation is uncertain, Local DP pins selected positions to box centers; a checked witness is a valid upper certificate, but restricted lower bounds never reject (Proposition 10). Methods run in separate processes with equal limits within a study and share no caches, certificates or reference solutions; generating trajectories are withheld.

Inputs. Designed studies D and E (6 and 12 inputs, $\beta = 2$) separate uncertain clusters by $s \in \{1, 2\}$ regular intervals, so that the minimum is either below the generator's support ($s = 1$) or certified structurally ($s = 2$). Log-derived queries integrate recorded arrival counts; these obey (1)–(3) exactly, because arrivals within a bin aggregate into one effective jump (Appendix I). We release some checkpoints exactly and others as quantization cells. Retail-I uses 8 one-hour windows of UCI Online Retail (Chen, 2015), and Retail-II 24 further windows from four other months, fixed before any solve; each window has six ten-minute labels and four release regimes, queried at all budgets $k = 0, \ldots, 6$. Taxi-I and Taxi-II each take 12 date/zone/hour windows of January 2025 NYC Green Taxi pickups (New York City Taxi and Limousine Commission, 2025), chosen by a rule fixed before inference, in exact and quantized regimes at $k = 3$. The timestamps are recorded; the uncertainty and the budget task are constructed.

6.1 Q1: decision coverage

Local DP decides every designed query (Table 1), including all eleven feasible budgets whose minimum lies below the generator's support, which only a witness can settle; capped enumeration leaves

Table 2: Taxi panels: 72 executions per method (24 cases, three correlated repeats). A/R/N/U: checked ACCEPT, exact REJECT, numerical negative only, UNKNOWN. CPU seconds and calls are three-repeat sums; FB counts enumeration fallbacks.

	Taxi-I							Taxi-II (unseen windows)						
Method	A	R	N	U	CPU	Calls	FB	A	R	N	U	CPU	Calls	FB
A: enumeration	36	36	0	0	31.09	1332	42	45	27	0	0	32.20	1407	54
B: Local DP	36	36	0	0	22.41	888	0	45	27	0	0	23.33	1095	0
C: MILP cascade	36	36	0	0	22.60	462	6	45	27	0	0	23.91	603	9
D: numerical MILP	36	0	36	0	17.86	108	0	45	0	27	0	17.91	117	0

eight unresolved. On the retail panels Local DP resolves 35 of the baseline's 36 UNKNOWNs in Retail-I and adds 126 decisions with no loss in Retail-II. Its remaining UNKNOWNs are budgets that conditional anchors cannot reject by design. With six labels, enumeration is affordable and decides every retail query, so these panels test soundness and near-completeness on log-derived shapes rather than speed.

6.2 Q2: COMPLETE COST ON TAXI PANELS

Methods A–C share Bounds + block-cap and continue with enumeration of supports of size at most three (A), Local DP with an enumeration fallback (B), or MILP with the same fallback (C); D runs the MILP alone. Each panel runs 24 cases $\times$ 4 methods $\times$ 3 repeats, with 30 CPU seconds, 60 wall seconds and 1,024 calls per execution and instrumentation qualified by eight controls. Complete CPU covers the controller and child processes from payload construction to certificate serialization.

A–C return identical certified answers on both panels (Table 2). Local DP never falls back to enumeration, uses 27.9% and 27.6% less complete CPU than enumeration, and is within 3% of the certifying MILP cascade. The numerical MILP is cheapest but cannot certify its negative answers. The Taxi-I cases had been used in two earlier runs (Appendix I.5), so only Taxi-II is unseen; repeats and regimes are correlated and the host was shared, so we infer no population speed ratio. At this size a certifying MILP cascade is a viable alternative, and our cost advantage is over enumeration.

6.3 Q3: LOCALIZATION AND INPUT LENGTH

Local versus full-prefix computation. A separate prospective probe removes preprocessing and runs both exact DP implementations directly on four fixed queries, crossing $\beta \in \{2, 4\}$ and $k \in \{3, 4\}$ at $n = 12$. Both methods agree on every decision, while local transfer LPs have fewer variables and constraints and require fewer calls in all four pairs (Table 3). Every bounded enumeration reference remains UNKNOWN at 128 calls. Analytic controls and a fresh zero-solver checker execution validate the retained DP outputs and reject certificate mutations (Appendix M).

Table 3: All four local/full-prefix pairs. Both DP methods return the same exact decision: R excludes every support through $k = 3$; E(4) certifies minimum four at $k = 4$. Dimensions are the maximum variables / constraints of the respective transfer or prefix LPs, not of the global witness LP. Calls include the global witness check. Every enumeration reference remains UNKNOWN at 128 calls.

		Both	LP dimensions		Calls		Reference
β	k	DPs	Local	Full-prefix	Local	Full-prefix	calls
2	3	R	10 / 32	22 / 68	20	22	128
2	4	E(4)	10 / 32	22 / 68	25	33	128
4	3	R	18 / 60	30 / 96	42	44	128
4	4	E(4)	18 / 60	30 / 96	75	87	128

The peak retained-record counts agree within every pair, so these results distinguish smaller projection problems from fewer observed states. The recurrence implementations differ; this is finite mechanism evidence and does not isolate reset merging or establish a general speed ratio.

Table 4: Heuristic event counts checked against certified minima on the retail inputs: exact / over-count / inconsistent (the count is below the minimum, or no compatible trajectory realizes the output). Trend filtering gives identical results for $\lambda \in \{0.01, 0.1\}$.

Proposer	Retail-I (32)	Retail-II (96)
ℓ_1 relaxation	8 / 24 / 0	8 / 88 / 0
Reweighted ℓ_1	24 / 8 / 0	58 / 38 / 0
Trend filtering on centers, $\lambda \leq 0.1$	8 / 10 / 14	1 / 43 / 52
Trend filtering on centers, $\lambda = 1$	0 / 0 / 32	6 / 4 / 86

Length and preprocessing. In the original Length study ($\beta = 2$, n from 9 to 66, 45 CPU seconds per method), every method was UNKNOWN for $n \geq 27$. Four later traces on one retained input locate the active stage at the optional all-triples packing bound. A matched ablation on two selected inputs shows that replacing this bound by the valid $\mathrm{OPT} \geq 0$ can let the continuation answer (Appendix L); this is not an attribution for every original timeout. In a separately checked post hoc rerun, Local DP settles all 16 original Length inputs up to $n = 66$, where enumeration settles four. On feasible large inputs the witness support matches the checked relaxation bound, giving exact minima 12 and 24, below the generator's 16 and 32. Its largest observed child CPU is 6.1 seconds, on different hardware from the original study, so the timing is indicative (Appendix Q). An additional prospective panel has an unresolved certificate-summary reconstruction mismatch and is retained only as a diagnostic, not confirmatory evidence (Appendix O).

Growing the number of blocks. A second prospective panel tests the structured regime with $\beta = 2$ held fixed while the number of uncertain clusters grows. Sixteen fresh inputs, with parameters fixed before any solve, have 4–32 uncertain clusters and $n = 15$–130. Each is queried at $k = 2b - 1$ using the typed zero packing bound, 45 child-CPU seconds, a 90-second wall watchdog and 512 calls. Local DP settles 14 of 16: all eight infeasible queries and six feasible ones, whose witness supports match checked lower bounds to give exact minima 6, 12 and 24. The two feasible $n = 99$ queries remain UNKNOWN at the call limit. Standalone MILP supplies witnesses for all eight feasible queries but only numerical negative statuses for the others; block-cap already rejects those eight structurally, while enumeration settles none. This separates the six additional closures beyond block-cap from inherited rejections. The local method's largest child CPU is 27.0 seconds on a heavily shared laptop; timing is indicative, not a clean-host comparison. All 80 attempts are retained in Appendix P.

6.4 Certifying heuristic event counts

Exact certificates can audit heuristics that propose sparse explanations. On the 128 retail inputs, whose minima are known, we ran the ℓ_1 relaxation (minimum total jump under all constraints), five rounds of reweighted ℓ_1, and ℓ_1 trend filtering of the box centers, which ignores the boxes (Table 4). The constrained proposers return valid upper bounds, but ℓ_1 over-counts on 112 inputs and reweighting on 46; since total jump equals net velocity change, ℓ_1 has many tied optima, one of which always attains the minimum, so its over-count reflects the solver's vertex. Trend filtering is inconsistent with the model on 66 inputs at small λ and on 118 at $\lambda = 1$. The certificate confirms each count, exhibits a smaller certified budget, or rejects the proposal, which no heuristic alone can do. This is a post hoc pilot on correlated, already analyzed inputs (Appendix R).

Mechanism and limits. Taxi witnesses do not reconstruct withheld midpoints within $1/100$ or compress the source series. In a downstream pilot fitting sparse convex surrogates to UCI Auto MPG (Quinlan, 1993), all 12 Local DP searches return UNKNOWN, while a witness-checked MILP saves storage within 1.05 times the dense test error in 10 of 12 contracts. A sensitivity study agrees with enumeration where implemented, but its block parameter equals n. Appendix G reports all of these.

7 DISCUSSION AND CONCLUSION

Exact minimum-event inference must preserve the association between each continuous state and a prefix that can reach it. The first-moment counterexample isolates a failure of that association, NP-completeness shows that the problem is hard without structure, and the shared-endpoint reset shows that exactly observed intervals erase support branching, confining it to local uncertainty blocks with controlled encoding length and exact witnesses. The output is a verifiable answer to a supplied model-complexity budget, which distinguishes an insufficient budget from a compatible sparse explanation even when the generating process used more events.

Limitations. The theorem identifies a sufficient class, not a characterization, and its parameter reaches n when uncertainty or timing restrictions connect the whole sequence; conditional anchors propose explanations for such inputs but cannot certify rejection. The experiments are finite and small: the log-derived queries have six labels, the taxi cost advantage is over enumeration and not over a certifying MILP cascade, the additional B3 panel has unresolved verification, the separate prospective Length panel is synthetic and reaches its call limit on two $n = 99$ inputs, the length rerun and the detector audit are post hoc on retained inputs, and the downstream pilot shows no Local DP benefit. The recorded logs supply event patterns but the uncertainty is constructed, so we claim neither latent-event recovery nor deployment value. The natural next steps are larger prospective panels with more uncertain samples and feasible budgets at large n, and certified audits of event counts proposed by learned detectors on data where the true events are known.

ETHICS STATEMENT

The experiments use UCI Online Retail and Auto MPG (both CC BY 4.0), and NYC TLC trip records, which are provider-submitted and distributed under NYC Open Data terms. The retail adapter exports only timestamps and aggregate counts, without invoice, customer or product identifiers, and the taxi adapter reads only pickup timestamps and pickup-zone identifiers. The log-derived tasks concern compatible event budgets, not individuals, demand forecasting or dispatch decisions. The Auto MPG pilot uses vehicle weight and fuel consumption, with approximate car-name groups for splitting; it is an exploratory surrogate task, not a deployment evaluation.

AI USE STATEMENT

Generative AI tools were used in model and hypothesis development, mathematical derivation and proof drafting, implementation, synthetic-input construction and recorded-data adaptation, experimental design, interpretation, literature search, internal checking, and manuscript preparation. Separate agent checks and deterministic rational verification were used for scoped mathematical and computational claims. These checks do not establish literature priority and do not replace the authors' responsibility for the manuscript. The computational experiments evaluate rational optimization procedures; they do not measure language-model performance.

REPRODUCIBILITY STATEMENT

The main text and appendices state the model, the relaxation, all proofs, the input-selection rules and input identities, every method's limits, and the interpretation of incomplete computations; Appendix G and the following appendices report every attempt of every study, including UNKNOWN and failed ones. For the comparisons used as certified evidence, each ACCEPT and REJECT has an exact rational certificate replayable without a solver; the unresolved panel in Appendix O is excluded. Method source, fixed rational inputs, outcome tables, figure generators and certificate checkers will be released; versioned raw outputs, input manifests and review records are retained, and the tables use the corrected final records.

REFERENCES

David L. Applegate, William Cook, Sanjeeb Dash, and Daniel G. Espinoza. Exact solutions to linear programming problems. *Operations Research Letters*, 35(6):693–699, 2007. doi: 10.1016/j.orl.2006.12.010.

Alberto Bemporad, Andrea Garulli, Simone Paoletti, and Antonio Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50 (10):1567–1580, 2005. doi: 10.1109/TAC.2005.856667.

Christoph Buchheim and Maximilian Merkert. Extended formulations for control languages defined by finite-state automata. *Mathematical Programming*, 2025. doi: 10.1007/s10107-025-02263-8. Published online 4 August 2025.

Daqing Chen. Online retail. UCI Machine Learning Repository, 2015. URL `https://archive.ics.uci.edu/dataset/352/online+retail`.

Kevin K. H. Cheung, Ambros Gleixner, and Daniel E. Steffy. Verifying integer programming results. In *Integer Programming and Combinatorial Optimization (IPCO 2017)*, Lecture Notes in Computer Science, pp. 148–160. Springer, 2017. doi: 10.1007/978-3-319-59250-3_13.

Michael Etscheid, Stefan Kratsch, Matthias Mnich, and Heiko Röglin. Polynomial kernels for weighted problems. In *Mathematical Foundations of Computer Science (MFCS 2015)*, Lecture Notes in Computer Science, pp. 287–298. Springer, 2015. doi: 10.1007/978-3-662-48054-0_24.

Paul Fearnhead, Robert Maidstone, and Adam Letchford. Detecting changes in slope with an L_0 penalty. *Journal of Computational and Graphical Statistics*, 28(2):265–275, 2019. doi: 10.1080/10618600.2018.1512868.

Samuel Fiorini and Kanstantsin Pashkovich. Uncapacitated flow-based extended formulations. *Mathematical Programming*, 153(1):117–131, 2015. doi: 10.1007/s10107-015-0862-9.

Ambros M. Gleixner, Daniel E. Steffy, and Kati Wolter. Iterative refinement for linear programming. *INFORMS Journal on Computing*, 28(3):449–464, 2016. doi: 10.1287/ijoc.2016.0692.

S. Louis Hakimi and Edward F. Schmeichel. Fitting polygonal functions to a set of points in the plane. *CVGIP: Graphical Models and Image Processing*, 53(2):132–136, 1991. doi: 10.1016/1049-9652(91)90056-P.

Toby Dylan Hocking, Guillem Rigaill, Paul Fearnhead, and Guillaume Bourque. Generalized functional pruning optimal partitioning (GFPOP) for constrained changepoint detection in genomic data. *Journal of Statistical Software*, 101(10), 2022. doi: 10.18637/jss.v101.i10.

Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi: 10.1006/jcss.2001.1774.

Rebecca Killick, Paul Fearnhead, and Idris A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012. doi: 10.1080/01621459.2012.737745.

Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009. doi: 10.1137/070690274.

Alessandro Magnani and Stephen P. Boyd. Convex piecewise-linear fitting. *Optimization and Engineering*, 10(1):1–17, 2009. doi: 10.1007/s11081-008-9045-3.

Robert Maidstone, Toby Hocking, Guillem Rigaill, and Paul Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27(2):519–533, 2017. doi: 10.1007/s11222-016-9636-3.

New York City Taxi and Limousine Commission. TLC trip record data. Official NYC TLC data page, 2025. URL `https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page`. January 2025 Green Taxi Trip Record; NYC Open Data/TLC terms; accessed 22 September 2026.

Joseph O'Rourke. An on-line algorithm for fitting straight lines between data ranges. *Communications of the ACM*, 24(9):574–578, 1981. doi: 10.1145/358746.358758.

Necmiye Ozay, Mario Sznaier, Constantino M. Lagoa, and Octavia I. Camps. A sparsification approach to set membership identification of switched affine systems. *IEEE Transactions on Automatic Control*, 57(3):634–648, 2012. doi: 10.1109/TAC.2011.2166295.

Quentin Ploussard. Piecewise linear approximation with minimum number of linear segments and minimum error: A fast approach to tighten and warm start the hierarchical mixed integer formulation. *European Journal of Operational Research*, 315(1):50–62, 2024. doi: 10.1016/j.ejor.2023.11.017.

Quentin Ploussard, Xiang Li, and Matija Pavičević. Tightening the difference-of-convex formulation for the piecewise linear approximation in general dimensions. *INFORMS Journal on Optimization*, 8(3):230–247, 2026. doi: 10.1287/ijoo.2025.0074.

R. Quinlan. Auto MPG [dataset]. UCI Machine Learning Repository, 1993. URL `https://archive.ics.uci.edu/dataset/9/auto+mpg`.

Ryan J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1), 2014. doi: 10.1214/13-AOS1189.

Juan Pablo Vielma. Embedding formulations and complexity for unions of polyhedra. *Management Science*, 64(10):4721–4734, 2018. doi: 10.1287/mnsc.2017.2856.

John Alasdair Warwicker and Steffen Rebennack. Efficient continuous piecewise linear regression for linearising univariate non-linear functions. *IISE Transactions*, 57(3):231–245, 2025. doi: 10.1080/24725854.2023.2299809.

Huanyu Zhao, Chaoyi Pang, Ramamohanarao Kotagiri, Christopher K. Pang, Ke Deng, Jian Yang, and Tongliang Li. An optimal online semi-connected PLA algorithm with maximum error bound. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):164–177, 2022. doi: 10.1109/TKDE.2020.2981319.

A THE FLOW ARCHITECTURE AND ITS EXACT-PREFIX COMPARATOR

This appendix fixes the relaxation used in Theorem 2; no conclusion about other flow formulations is implied. Layers are $i = 0, \dots, n$. A node is (i, c, H), with H the ordered subset of designated labels strictly less than i and $c = |H| \le k$. Across interval i, an inactive arc leaves H unchanged and an active arc appends i. Active means *allowed*, not required, to have a positive jump. Multiple final nodes feed a super-sink.

An arc e has mass $f_e \ge 0$ and scaled state variables $P_e^-, W_e^-, P_e^+, W_e^+, J_e, R_e$. Write $J_{\max} = J$ to distinguish the cap from J_e. Its constraints are

$$P_e^+ - P_e^- = DW_e^- + R_e, \qquad W_e^+ - W_e^- = J_e, \tag{11}$$

$$0 \le J_e \le J_{\max} f_e, \qquad (D - u_i)J_e \le R_e \le (D - l_i)J_e, \tag{12}$$

$$a_i f_e \le P_e^- \le b_i f_e, \qquad a_{i+1} f_e \le P_e^+ \le b_{i+1} f_e. \tag{13}$$

Inactive arcs impose $J_e = R_e = 0$. To make zero-mass states unambiguous, impose $-Mf_e \le W_e^{\pm} \le Mf_e$, where

$$M = (\max_i b_i - \min_i a_i)/D + J_{\max}.$$

This is a valid bound for every physical trajectory: each w_i, $i < n$, follows from its displacement equation and $0 \le r_i \le DJ_{\max}$; $w_n = w_{n-1} + j_{n-1}$ obeys the same symmetric bound. Stronger valid bounds are optional and are not needed for the stated construction.

The source emits unit mass. At every other nonsink node v, require

$$\sum_{e \in \delta^-(v)} f_e = \sum_{e \in \delta^+(v)} f_e, \quad \sum_{e \in \delta^-(v)} P_e^+ = \sum_{e \in \delta^+(v)} P_e^-, \quad \sum_{e \in \delta^-(v)} W_e^+ = \sum_{e \in \delta^+(v)} W_e^-. \tag{14}$$

There is no common unscaled state shared by all incident arcs.

For every head or tail copy at sample i and count c, of mass f, attach its *own* future variables $\widehat{P}_t, \widehat{W}_t, \widehat{J}_t, \widehat{R}_t, Z_t$. Fix its initial scaled state to that copy; impose the scaled observation and dynamical constraints for $t = i, \dots, n-1$, velocity bounds $\pm Mf$, and

$$0 \le Z_t \le f, \qquad 0 \le \widehat{J}_t \le J_{\max} Z_t, \qquad \sum_{t=i}^{n-1} Z_t \le (k - c)f. \tag{15}$$

For a head after an active arc, c includes that arc. For a tail before an active arc, it does not. These independent futures need not agree with the outgoing arc decomposition. This is the precise full-horizon fractional-activation lookahead in $\mathcal{F}_k$.

For the comparator in Proposition 3, let $Q_{i,H}$ be the complete original prefix polyhedron through sample i, with jumps zero outside H. At each outgoing tail attach a separate mass-scaled copy of $Q_{i,H}$, identifying its final position and velocity with that tail. A positive final arc then has an actual allowed-mask prefix and one actual final step. Their shared endpoint gives a complete physical witness with at most k positive labels. This argument does not assert that all positive-flow arcs can be glued into mutually compatible trajectories, nor that the entire augmented feasible region is an ideal convex-hull description. There can be $\sum_{c=0}^{k} \binom{i}{c}$ histories at layer i.

B A RATIONAL FAMILY WITH LOST WITHIN-HISTORY CORRELATION

B.1 THE PHYSICAL MINIMUM

For arbitrary duration D, reversal is

$$p'_i = p_{n-i}, \quad w'_i = -w_{n-i}, \quad j'_i = j_{n-1-i}, \quad r'_i = Dj_{n-1-i} - r_{n-1-i}.$$

It preserves support, nonnegative jumps and their cap, and maps a window $[l, u]$ to $[D - u, D - l]$. Free endpoint velocities introduce no extra condition. Midpoint windows are unchanged.

Reverse the family in (4). Its first four positions are $0, 0, 1/4, 7/8$. With $D = 1$ and $r_i = j_i/2$, consecutive displacement differences give $j_0 + j_1 = 1/2$ and $j_1 + j_2 = 3/4$. Thus

$$(j_0, j_1, j_2) = (t, 1/2 - t, 1/4 + t), \qquad 0 \le t \le 1/2, \tag{16}$$

$$(w_0, w_1, w_2, w_3) = (-t/2, t/2, 1/2 - t/2, 3/4 + t/2). \tag{17}$$

For $0 < t < 1/2$, this already requires three positive labels. At $t = 0$ or $1/2$, it requires two, so a total budget of two forces $j_3 = j_4 = 0$. The last two positions are then $13/8+t/2$ and $19/8+t$. For $0 < \lambda < 1$, the required final value $19/8 + \lambda/2$ matches neither endpoint choice. A budget of one is impossible even in this prefix. Taking $t = \lambda/2$ and a zero suffix supplies a compatible three-label trajectory. At $\lambda = 0, 1$, the same construction has two labels and proves both endpoint statements.

In the original orientation, an explicit upper witness is

$$w_0 = -3/4 - \lambda/4, \qquad (j_0, \dots, j_4) = (0, 0, 1/4 + \lambda/2, (1-\lambda)/2, \lambda/2), \qquad r_i = j_i/2. \quad (18)$$

It has $p_1 = 13/8 + \lambda/4$, which belongs to the declared observation interval, and all jumps lie in $[0, 1]$.

B.2 The budget-two relaxed witness

Use that physical witness through interval two, with unit mass and designated history (2). At sample three the state is $(p, w) = (1/4, -1/2 + \lambda/4)$. Replace its single outgoing state by these two normalized tails:

Mass	w_3	j_3	w_4	j_4	w_5
$1-\lambda$	$-1/2$	$1/2$	0	0	0
λ	$-1/4$	0	$-1/4$	$1/2$	$1/4$

Both tails have $p_3 = 1/4, p_4 = p_5 = 0$ and $r_i = j_i/2$. Their histories become $(2, 3)$ and $(2, 4)$, respectively. Their incoming position moments agree, and

$$(1-\lambda)(-1/2) + \lambda(-1/4) = -1/2 + \lambda/4.$$

Thus all flow constraints hold. No distinct incoming histories are merged at the split.

For common-prefix copies, use the remaining portion of the displayed physical witness in their fractional lookahead, with $z_i = j_i$ since $J = 1$. Before the designated event in slot two, total remaining activation is $3/4+\lambda/2 < 2$. Immediately after it, remaining activation is $j_3+j_4 = 1/2 \leq 1$. Earlier copies only remove zero jumps. Each split tail has a physical one-event suffix, admissible to its fractional lookahead; after its second designated event the remaining suffix is inactive. These witnesses cover every nonzero head and tail copy and satisfy the displayed finite velocity bound.

For completeness, the normalized lookahead budgets at all copy types are:

Copy type	History count c	Remaining activation	Allowed $2 - c$
Common prefix before event 2	0	$3/4 + \lambda/2$	2
Common head after event 2	1	$1/2$	1
Either split branch, before its second event	1	$1/2$	1
Either branch, after its second event	2	0	0

Inactive arcs only advance these same suffix witnesses. On a copy of mass h, multiply the displayed physical and activation variables by h; its activation bound is $h(2 - c)$. In particular, split tails at sample three still have $c = 1$; $c = 2$ applies only after their events in slots three or four.

For the exact history (2), the fixed prefix observations force the unique velocity $-1/2 + \lambda/4$ at sample three: before slot two velocity is constant, while the exact positions at samples zero, two and three determine it and the only jump. The two outgoing residuals relative to this forced velocity are $-\lambda/4$ and $(1 - \lambda)/4$. They are nonzero and cancel only after weighting. Exact-prefix copies therefore reject the artificial split. The result concerns feasibility at budget two; no assertion about the smallest relaxed feasible budget is needed.

C Proof of the reset/block algorithm

Remarks on the reset. For $u < D$, g_u increases up to the crossing point $s - J(D - u)/D$ of its two affine pieces and does not increase afterwards, so the maximum in (8) is attained at that point if

it lies in H and otherwise at an endpoint of a component of H. For $u = D$ only the largest feasible incoming velocity is needed, and for $u = 0$ the image is $\{s\}$. The degenerate window $l_i = u_i = D$ forces $w = s$, so failure of the reset condition does not by itself imply intractability. Initial and final blocks keep their original boundary boxes, and the final block needs only a feasibility test. Figure 3 illustrates the block structure.

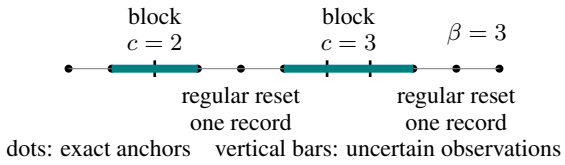

Figure 3: Block structure on a ten-interval example. Uncertain sample 2 induces a two-interval block, and samples 6 and 7 induce a three-interval block. A block of length c can produce 2^c records per budget; the following regular interval, whose timing window opens at zero, compresses their exact union to one record. The cost depends on the longest block, not on the number of uncertain samples.

Algorithm summary. Maintain one array indexed by $q = 0, \ldots, \bar{k}$, whose entries are exact unions of interval records, with masks/backpointers.

1. Compute $\mathcal{U}$, $\mathcal{B}$ and their maximal blocks from the observation boxes. Check the regular-window condition.
2. Process a first regular interval with free incoming velocity, or enumerate the masks of an initial block with its original boundary boxes.
3. On each subsequent block, for every budget enumerate its allowed masks and project the corresponding block LP with the exact predecessor interval, as in (10).
4. On each regular interval, apply (9) and keep the predecessor attaining the largest right endpoint, or an inactive witness if it is the only feasible branch.
5. Accept exactly the nonempty terminal budgets. Recover a selected global mask through the stored pointers and solve one masked LP for a physical witness. The least accepted budget is the minimum when budgets through n are processed.

These are exact rational operations. An implementation that uses numerical proposals must return UNKNOWN whenever it cannot recover the required exact certificates.

C.1 The regular transition, including budgets

For a regular interval, substitute $r = D(s - w)$ into the timing inequalities with $l = 0$. For $u < D$ this gives

$$s - w \le j \le \min\{J, D(s - w)/(D - u)\}, \qquad s - J \le w \le s.$$

For $u = D$, the upper limit on j is simply J. Adding w proves (7), including $J = 0$. Compactness gives Lemma 5.

Let $V_i(q)$ be the exact reachable incoming velocity set with at most q positive labels, and set $V_i(-1) = \varnothing$. The next set is the union of the inactive point s, if $s \in V_i(q)$, and the allowed-jump image of $V_i(q - 1)$. The latter is charged one label but permits $j = 0$. This never creates a trajectory with more than q positive events. Conversely every trajectory with a positive current jump is included, while a zero current jump is in the inactive branch. Both branches share the endpoint s, so their union is empty or one closed interval. This specifies the budget recurrence without requiring strictly positive LP variables.

Initially velocity is free. If interval zero is regular, intersect the free line with $[s - J, s]$ for the charged transition; the inactive branch is the feasible point s. After this first transition all recorded intervals are bounded. An initial uncertainty block is solved directly with its original initial position box and free initial velocity.

C.2 EXACT BLOCK TRANSFERS

For a maximal block $C = \{a, \dots, b\}$ of length c, every internal cut outside it has an exact position. In particular, p_a is exact unless $a = 0$, and p_{b+1} is exact unless $b = n - 1$: otherwise an adjacent interval would also belong to $\mathcal{B}$. Hence a transfer can retain only velocity at an internal boundary without dropping position correlation.

Fix an incoming interval and an allowed local mask $A \subseteq C$. The local LP contains $c + 1$ positions and velocities, c jumps and moments, all original boxes and window inequalities, and the incoming velocity bound. It has $O(c)$ variables and constraints. For $c \geq 1$, all its positions are bounded and every velocity is bounded by a displacement equation, a finite jump cap and the final update. It is therefore compact. Its projection onto the last velocity is either empty or an attained rational interval, obtainable from feasibility and two endpoint optimizations.

If the block starts the instance, use its initial box and no predecessor bound. For a terminal block, feasibility suffices. If it occupies the whole instance, then $n \leq \beta$ and direct enumeration of its masks has the asserted bound. Consecutive maximal blocks are separated by at least one regular interval. Each block consequently receives at most one interval per predecessor budget and produces at most one interval per mask. Formula (10) proves the 2^β record bound.

To prove completeness, restrict any physical prefix to its positive local labels and its incoming velocity; it appears in the corresponding transfer. To prove soundness, a feasible local point has an incoming velocity in an exact predecessor projection. Choose a predecessor witness for that velocity. Their boundary positions are the same exact number, so concatenate them. The resulting support is contained in the union of the charged masks. Boundary blocks are sound by their complete original boxes. This induction also proves that no intermediate union has been convexified without equality.

C.3 ORIGINAL-COEFFICIENT PREFIX INVARIANT AND BIT SIZE

The following stronger invariant prevents a circular bit-complexity argument: every stored interval is the projection of a full original-coefficient prefix LP for one allowed mask of size at most its budget. At a block, concatenate the predecessor mask with A; identifying an exact boundary position and ranging over its predecessor velocity interval is precisely projection of this combined LP.

At a regular transition, each predecessor record produces an interval beginning at s. Among the finitely many records, one has the largest upper endpoint. Its extended mask represents the entire charged union, and also contains any feasible inactive point. If the charged union is empty but the inactive point is feasible, choose a predecessor mask witnessing that point and append an inactive interval. The selected mask thus represents the whole resulting interval. At the initial regular step the assertion is immediate from its one-step LP.

Each such full-prefix system has $O(n)$ variables and constraints, using original rational coefficients and forced-zero equalities. A nonempty prefix containing at least one interval is bounded: position boxes and jump caps bound all positions and jumps; the first displacement equation bounds its incoming velocity, and the velocity recurrence bounds the rest. Hence every finite endpoint is attained at a vertex. Clear denominators in each row. The resulting integer coefficients have bit length polynomial in L; for an $m \times m$ vertex basis with $m = O(n)$ and entry bit length B, Hadamard's bound gives determinant bit length $O(m(B + \log m))$. Cramer's rule therefore bounds every coordinate of a vertex, and every retained endpoint, by $\mathrm{poly}(n, L)$ bits. A zero-length free boundary is kept symbolically rather than asserted to have a bounded vertex.

The locally passed endpoints inherit this original-prefix bound; they are not a fresh uncontrolled recursive source of coefficient growth. Rational LP on each $O(\beta)$ block system, plus regular affine operations, thus costs polynomial bit time per record. There are $O(n(\bar{k} + 1)2^\beta)$ endpoint or interval operations, proving Theorem 6. This is a Turing-polynomial factor, not a strongly polynomial claim.

Store the selected predecessor and local mask with each retained record. Once a terminal budget is feasible, these pointers yield an allowed global mask. One global masked LP recovers a rational vertex witness with the same polynomial encoding bound. Division in $\tau_i = D - r_i/j_i$ adds only polynomially many bits for positive j_i, even when that rational jump is small; inactive labels need no event time. Trying all budgets through n, or recording them in one forward pass, yields the

minimum-support bound. Empty final sets for every budget mean infeasibility, not an unknown optimum.

Corollary 8 (Isolated uncertainty)**.** *If all windows are $[0, D]$ and distinct uncertain sample indices differ by at least three, then $\beta \leq 2$. The algorithm uses at most four records per budget at a block exit and one after a regular interval, giving polynomial-time minimum-support inference.*

Proof. Each uncertain sample touches at most its two incident intervals. Separation by at least three leaves a regular interval between successive components. All regular lower window endpoints are zero, so Theorem 6 applies. □

C.4 Absorbing local violations of the reset condition

Corollary 9 (Joint position-and-timing blocks)**.** *For arbitrary valid timing windows, form $\mathcal{B}^+ = \mathcal{B} \cup \{i : l_i > 0\}$ and let β^+ be its largest component size, or zero if empty. The reset/block algorithm, using these enlarged blocks, decides support at most k with at most 2^{β^+} records per budget at a block exit and one after each remaining regular interval. Its Turing complexity is $2^{\beta^+}\,\mathrm{poly}(n, \bar{k}, \beta^+, L)$; minimum support is computable in $2^{\beta^+}\,\mathrm{poly}(n, \beta^+, L)$ time.*

Proof. Every interval outside $\mathcal{B}^+$ has exact position endpoints and lower timing endpoint zero. The entry and exit positions of an enlarged block are exact unless they are p_0 or p_n: if such a position were uncertain, both of its incident intervals would belong to $\mathcal{B} \subseteq \mathcal{B}^+$, contradicting maximality of the component. Each enlarged block therefore admits the same complete local masked LP projection and exact concatenation argument as above, with arbitrary valid windows inside it. Distinct blocks are separated by a shared-endpoint reset. Replace the old block lengths by their enlarged lengths in the record-count induction. The original-coefficient prefix invariant and bit-size bound are unchanged. Initial blocks, terminal blocks, an all-block instance and an empty block set retain their earlier boundary treatments. □

The corollary extends the same mechanism; it is a sufficient condition, not a classification. A single timing-restricted interval can join neighboring uncertainty components, so its cost cannot be inferred merely from the number of exceptional intervals. If every interval belongs to the enlarged set, the bound is exponential in n.

D NP-completeness of the budget query

This appendix proves Proposition 4. Throughout, $D = J = 1$, and $s_i = p_{i+1} - p_i$ denotes the displacement over interval i. With delay τ_i, (1) gives

$$s_i = w_i + (1 - \tau_i) j_i, \qquad s_{i+1} - s_i = \tau_i j_i + (1 - \tau_{i+1}) j_{i+1}. \tag{19}$$

Membership. A certificate is a support S with $|S| \leq k$. The masked LP of Section 3 decides in polynomial time whether a physical trajectory supported within S exists.

Source problem. In exact-cardinality subset sum we are given nonnegative integers $a_0, \ldots, a_{m-1}$ with $m \geq 1$, a target $A \geq 0$ and a cardinality $0 \leq R \leq m$, and ask for $Q \subseteq \{0, \ldots, m-1\}$ with $|Q| = R$ and $\sum_{h \in Q} a_h = A$. This is NP-hard: an instance of ordinary subset sum with d items becomes one with $2d$ items after appending d zeros and setting $R = d$.

Construction. Let $N = 2m + 4$ intervals, $e = 2m + 2$ and budget $k = m$. Interval $i(h, b) = 2 + 2h + b$, for $h < m$ and $b \in \{0, 1\}$, is a *candidate*; intervals $0, 1, e, e + 1$ are *boundary* intervals. Set

$$C = 1 + \textstyle\sum_h a_h + A, \quad T_h = C\, 3^h, \quad \Omega = \max_h T_h + \max_h a_h + 1, \quad \eta = \tfrac{1}{16 m \Omega},$$

$$\zeta = \frac{\eta}{8m}, \quad \epsilon = \frac{\eta}{8(6e+2)}, \quad P = \textstyle\sum_h \big(2m - 2h - \tfrac{1}{2}\big) - R + \eta\big(\sum_h T_h + A\big), \quad M = mN + 1.$$

Candidate $i(h, b)$ has nominal delay $\tau_i^0 = 1/2 - \eta(T_h + ba_h)$, boundary intervals have $\tau_i^0 = 1/2$, and every window is $[\tau_i^0 - \zeta, \tau_i^0 + \zeta]$. Since $\eta(T_h + ba_h) < \eta\Omega \le 1/16$ and $\zeta \le 1/128$, every window has positive width and lies in $(1/4, 3/4)$. The boxes for $p_0, p_1, p_2, p_e, p_{e+1}, p_{e+2}$ are centered at $0, 0, 0, P, P + m, P + 2m$ with half-width ϵ, and every other box is $[-M, M]$. All numbers have encoding length polynomial in the input.

Step 1: boundary jumps are small. The first three boxes give $|s_0|, |s_1| \le 2\epsilon$, so $0 \le s_1 - s_0 \le 4\epsilon$. Both coefficients in (19) are at least $1/4$, hence $j_0 + j_1 \le 16\epsilon$. The last three boxes give $j_e + j_{e+1} \le 16\epsilon$ in the same way.

Step 2: the budget saturates and excludes boundary jumps. By (19), $w_0 \le s_0 \le 2\epsilon$ and $w_N = s_{N-1} + \tau_{N-1} j_{N-1} \ge s_{N-1} \ge m - 2\epsilon$. The total jump $\Gamma = w_N - w_0$ therefore satisfies $m - 4\epsilon \le \Gamma \le m$, where the upper bound uses at most m positive unit-capped jumps. Since $4\epsilon < 1$, exactly m labels are positive, and each positive jump is at least $\Gamma - (m-1) \ge 1 - 4\epsilon > 16\epsilon$. By Step 1, no boundary label is positive, so the m positive labels are candidates and $\sum_{\text{active}}(1 - j_i) = m - \Gamma \le 4\epsilon$.

Step 3: exact decoding of the nominal sum. With boundary labels inactive, $w_0 = s_0$ and $|w_0| \le 2\epsilon$. Every candidate jump occurs before time e, so

$$p_e = p_0 + e\, w_0 + \textstyle\sum_{\text{active}}(e - i - \tau_i) j_i,$$

and the boxes give $|\sum_{\text{active}}(e - i - \tau_i) j_i - P| \le (2e + 2)\epsilon$. Let $W = \sum_{\text{active}}(e - i - \tau_i^0)$. Each term changes by at most $e(1 - j_i) + \zeta$ when j_i is replaced by one and τ_i by τ_i^0, so

$$|W - P| \le (2e + 2)\epsilon + 4e\epsilon + m\zeta = \eta/4.$$

For candidate (h, b), $e - i - \tau_i^0 = 2m - 2h - b - \frac{1}{2} + \eta(T_h + ba_h)$. Both W and P therefore lie in $\mathbb{Z} - \frac{m}{2} + \eta\mathbb{Z}$, so $W - P \in \eta\mathbb{Z}$ because $1/\eta$ is an integer. Hence $W = P$.

Step 4: recovery of a subset-sum solution. Let $c_h \in \{0, 1, 2\}$ be the number of active candidates in pair h, and let $t_h = 1$ if candidate $(h, 1)$ is active. Then $W - P = Z + \eta X$ with $Z \in \mathbb{Z}$ and

$$X = C \textstyle\sum_h (c_h - 1) 3^h + \sum_h t_h a_h - A.$$

The active encoded terms sum to less than $m\eta\Omega = 1/16$, and $\eta(\sum_h T_h + A) < (m+1)\eta\Omega \le 1/8$, so $|\eta X| < 1$. Since $Z = -\eta X$ is an integer, $Z = X = 0$. Because $|\sum_h t_h a_h - A| < C$, reducing X modulo C gives $\sum_h t_h a_h = A$ and $\sum_h (c_h - 1) 3^h = 0$. The coefficients lie in $\{-1, 0, 1\}$, and uniqueness of balanced ternary representations forces $c_h = 1$ for every h. Then $Z = R - \sum_h t_h = 0$, so $\{h : t_h = 1\}$ has size R and sum A.

Step 5: converse. Given a solution Q, activate candidate $(h, [h \in Q])$ in each pair with a unit jump at its nominal delay, set $p_0 = w_0 = 0$, and leave the boundary labels inactive. Then $p_1 = p_2 = 0$ and $p_e = W = P$, since $Z = X = 0$. The velocity after the last candidate is m, so $p_{e+1} = P + m$ and $p_{e+2} = P + 2m$. All other positions lie in $[0, mN] \subseteq [-M, M]$. This physical trajectory has support $m = k$.

Steps 1–5 give a polynomial many-one reduction, which proves Proposition 4. The widths ϵ and ζ shrink with the encoded numbers, so the result is weak NP-hardness. It gives no hardness for fixed noise levels, for wide common windows, for fixed budgets (supports of size at most a fixed k can be enumerated in polynomial time), or for bounded block length. Every interval of the construction touches an uncertain sample, so $\beta = \beta^+ = N$; condition (6) holds vacuously because there are no regular intervals.

Consequence under ETH (Corollary 7). Start from ordinary Subset Sum with d numbers. Padding with d zeros gives exact-cardinality subset sum with $m = 2d$ items and $R = d$, and the construction above gives $N = 2m + 4 = 4d + 4$ intervals, $\beta = \beta^+ = N$, and encoding length polynomial in the Subset Sum input. Subset Sum admits no $2^{o(d)}$-time algorithm unless ETH fails; the standard proof reduces 3-SAT with v variables and c clauses to Subset Sum with $2v + 2c$ numbers of polynomial bit length and applies the sparsification lemma (Impagliazzo et al., 2001; Etscheid et al., 2015). On those instances an algorithm running in $2^{o(\beta)}\,\mathrm{poly}(n, L)$ time for the budget query would run in $2^{o(d)}$ time, a contradiction. Because every constructed instance satisfies (6), the 2^{β} factor of Theorem 6 cannot be improved to $2^{o(\beta)}$ on the class it covers.

E THE LOCAL-SEPARATOR COMPARISON METHOD

For completeness, this section gives an additional separator algorithm for more general regular windows. It is distinct from the reset/block DP tested in Section 6. It uses the same lifted model as Section 3, without requiring zero lower timing endpoints. Let $R = \{0, \dots, n-1\} \setminus \mathcal{B}$ be the regular intervals. Build a directed acyclic graph whose ordered vertices are a free source -1, every interval in R, and a free sink n. A real vertex designates its interval as inactive. Since its two position endpoints are exact, inactivity fixes both velocities to its chord slope.

For vertices $a < z$, let $R(a, z)$ and $S(a, z)$ be the regular and special intervals strictly between them, and write $r = |R(a, z)|$. Retain the endpoint pair only if $r \le \bar{k}$. For every $A \subseteq S(a, z)$ with $|A| \le \bar{k} - r$, test the complete original block LP with special labels outside A forced inactive and all interior regular labels allowed. At a real left separator, the incoming block state is its exact right position and pinned chord velocity; at a real right separator, the outgoing block state is its exact left position and pinned velocity. Free boundaries retain their original boxes and free velocity. The separator intervals themselves are bridges and need not be duplicated in the block LP. Adjacent real separators are compatible precisely when their pinned velocities agree.

A feasible test supplies an edge (a, z, A) of weight $r + |A|$. A source–sink path of weight at most $\bar{k}$ implies a physical trajectory: concatenate feasible block witnesses through the exact pinned separator states. Allowed labels can have zero jumps, so physical support is no greater than path weight. Conversely, from a physical trajectory of support $q \le \bar{k}$, take *every* inactive regular interval as a path vertex. Every skipped regular interval is then positive; choose A to be its positive special labels. Each edge was enumerated, and the resulting path has weight exactly q. This proves feasibility equivalence and equality of the shortest path weight with the minimum support whenever that support is at most the supplied cap.

There are at most $r + 1$ special components between an endpoint pair: distinct components require a regular interval between them. Hence $|S(a, z)| \le (r + 1)\beta \le (\bar{k} + 1)\beta$. There are $O(n(\bar{k} + 1))$ eligible endpoint pairs and at most $2^{(\bar{k}+1)\beta}$ local masks per pair. Exact rational LP and a shortest-path pass give

$$2^{(\bar{k}+1)\beta} \operatorname{poly}(n, \bar{k}, \beta, L)$$

bit complexity. For any fixed left vertex, an eligible right endpoint is among the next $\bar{k} + 1$ regular vertices, with at most one additional sink choice; long special runs introduce no additional vertices. This is a parameterized guarantee in (k, β), not in β alone for unrestricted optimum inference. If no real separator exists, the source–sink test retains the entire original instance; when special intervals are present they form one component, so its length is at most β. The construction also covers $k = 0$, $J = 0$, singleton windows and globally infeasible observations.

For an executable lower certificate, testing only some edge states is insufficient. A decisive negative answer requires a complete fixed schedule, a checked infeasibility certificate for every absent feasible possibility needed by that schedule, and a verified absence of a within-budget path. Missing, duplicated or unresolved states cannot be treated as excluded. A complete replay must enforce this boundary. A physical upper witness and exclusion of the immediately smaller budget together certify its reported optimum.

F EXTENDED RELATED WORK

This appendix expands Section 2.

Changepoint detection and trend filtering. Exact dynamic programming for segmentation is classical in time-series analysis. PELT and functional pruning compute penalized optimal changepoint segmentations exactly with pruning (Killick et al., 2012; Maidstone et al., 2017), and GFPOP adds constraints between consecutive segment parameters (Hocking et al., 2022). CPOP optimizes a continuous piecewise-linear fit with squared-error loss and an L_0 penalty on changes in slope, conditioning on a continuous endpoint value and pruning candidate cost functions (Fearnhead et al., 2019). ℓ_1 trend filtering is the convex relaxation of penalizing slope changes (Kim et al., 2009; Tibshirani, 2014). Continuous-state dynamic programming for slope changes is therefore established. These methods optimize a statistical fit criterion. We instead answer a hard-feasibility budget query

under capped nonnegative jumps and timing windows, with certificates for both answers. Our technical point is not a continuous DP state but an exact collapse of its feasible-set union at regular intervals.

Bounded-error and set-membership identification. Bounded-error identification seeks the fewest affine submodels, or the fewest switches, consistent with measurements whose errors lie in known intervals (Bemporad et al., 2005; Ozay et al., 2012). Ozay et al. (2012) relax the number of parameter changes by ℓ_1 sparsification. Our observation boxes play the same role as bounded errors, but the dynamics are fixed, and we seek an exact certified minimum under jump caps and labelled timing windows rather than a greedy or convex surrogate.

Piecewise-linear fitting within tolerances. Lines and polygonal functions can be fit to data within vertical tolerances by efficient geometric algorithms (O'Rourke, 1981; Hakimi & Schmeichel, 1991). Zhao et al. (2022) minimize the number of semi-connected segments under a maximum-error bound in linear time, and Ploussard (2024) give a hierarchical mixed-integer formulation that first minimizes the number of continuous segments and then the approximation error. Warwicker & Rebennack (2025) give continuous univariate approximation results that preserve convexity, with a runtime guarantee under smoothness and known convexity-change assumptions. Magnani & Boyd (2009) fit convex piecewise-linear functions, including fixed-size max-affine representations. Our nonnegative jumps also produce a convex detrended trajectory, so continuity and convexity alone do not distinguish our setting. The distinguishing constraints are capped jumps, per-label timing windows and label-counted support. Our NP-completeness reduction (Proposition 4) encodes its numbers in the timing windows. None of these works states the shared-endpoint reset under the joint cap, label and timing constraints, and we make no priority claim for minimum-segment bounded-error fitting itself.

Disjunctive, automaton and flow formulations. For a fixed allowed support the lifted feasible set is polyhedral, so the problem is a union of support-specific polyhedra, which is naturally related to embedding formulations (Vielma, 2018). Our exact-prefix comparator uses this standard disaggregation principle. Recent general-dimensional PLA work tightens a difference-of-convex MILP through a well-behaved interpolation representation (Ploussard et al., 2026). Its affine-piece adjustments preserve sampled values, not our fixed jump labels, windows and cap, whereas our reset preserves the complete feasible velocity image under those constraints. Regular-language polytopes admit flow-based extended formulations derived from finite automata (Fiorini & Pashkovich, 2015), and Buchheim & Merkert (2025) extend automaton-based descriptions to continuous control languages in function space. Our negative result does not contradict these constructions. It concerns a weaker architecture that conserves first continuous-state moments while giving outgoing copies independent suffix lookaheads. In that architecture a complete label history is not the same object as a complete continuous disjunct.

Exact certification. We follow the standard pattern of a floating-point proposal followed by exact rational verification (Applegate et al., 2007; Gleixner et al., 2016). Our REJECT answers carry replayable exact lower certificates: Farkas certificates of infeasibility or rational structural bounds. We do not certify MILP infeasibility itself, which would require verified branch-and-bound certificates (Cheung et al., 2017). For this reason, numerical MILP exclusions are reported at their own evidence level.

Table 5 summarizes these objectives. Our empirical comparisons concern implementations within the stated model, not superiority over methods designed for other tasks.

Table 5: Problem-level comparison of objectives and guarantees. None of these formulations supplies a matched runtime benchmark for our joint hard constraints.

Method	Objective and constraints	Representation or guarantee
CPOP (Fearnhead et al., 2019)	Continuous fit; squared error plus L_0 slope-change penalty	Continuous endpoint DP with exact pruning
ℓ_1 trend filtering (Kim et al., 2009)	Squared error plus ℓ_1 penalty on second differences	Convex relaxation of sparse slope changes
Switched affine SM-ID (Ozay et al., 2012)	Fewest parameter switches under bounded errors	ℓ_1 sparsification
Semi-connected PLA (Zhao et al., 2022)	Fewest semi-connected segments; maximum-error bound	Optimal online linear-time algorithm for its model
Hierarchical PLA (Ploussard, 2024)	Fewest continuous segments, then smallest error, under an error bound	Hierarchical mixed-integer formulation with tightening and warm starts
This work	Fewest positive labelled jumps; hard boxes, nonnegative cap, timing windows	Exact feasible-velocity sets; NP-complete in general; $2^{\beta}\,\mathrm{poly}(n, \beta, L)$ on the stated class

G EXTENDED EXPERIMENTAL REPORT

This appendix reports every study summarized in Section 6, with complete denominators, displays and verification boundaries. The taxi cost table appears in the main text (main-text Table 2).

We ask three empirical questions. (Q1) Does exact local inference settle budget queries that cheaper bounds leave open, on designed inputs and on queries built from recorded timestamps? (Q2) When all certifying methods reach the same answers, what does a complete certified decision cost relative to exhaustive enumeration and a certifying MILP cascade? (Q3) Where does the current implementation stop working? The budget studies are not forecasting or dispatch benchmarks. A separate exploratory regression-surrogate study asks whether a sparse compatible function can reduce storage while preserving predictive error; we report it separately because its task and guarantees differ.

G.1 TASK, METHODS AND EVIDENCE MAP

A query supplies observations and a budget k. A method returns ACCEPT when it constructs a physical witness with at most k positive labels, REJECT when an original-model lower certificate exceeds k, and UNKNOWN otherwise, including when a resource limit is reached. UNKNOWN therefore records incomplete computation, not infeasibility. A method can settle a budget without proving the exact minimum. The number of events used to generate an input is never used as a label.

Methods. Except for the standalone MILP, every method starts from the same kind of front end and extends it differently.

- **Bounds** takes the larger of an observation-derived packing lower bound and the lower bound of the LP relaxation of the activation formulation. A relaxation solution is kept as an upper witness when it passes the exact original-model check.
- **Block-cap** adds a structural lower bound: exact affine triples fix boundary velocities, and the jump cap bounds how many events a velocity change needs (Appendix H).
- **Local DP** continues with the reset/block algorithm of Section 5, or its joint-block version.
- **Enumeration** continues with exhaustive enumeration of original-model supports, each settled by an exact masked-LP certificate.
- **MILP** runs either after the front end or standalone. It solves a feasibility MILP with a binary activation z_i per label, $0 \le j_i \le Jz_i$, $\sum_i z_i \le k$, and the original boxes, dynamics and windows (SciPy/HiGHS). A positive answer is accepted only after its witness passes the exact check. A numerical infeasibility status is not an exact exclusion.
- **Conditional anchors** propose witnesses when the original blocks are long, for example when every observation is uncertain. They pin selected positions to their box centers, solve the restricted small-block problem, and check any witness against the original boxes (Proposition 10). Restricted lower bounds are never used to reject.

Each method runs in its own process, computes its own front end, and shares no certificates, caches or reference solutions with other methods. In the designed and recorded-log budget studies, source values and generating trajectories are withheld from every method. The regression-surrogate experiment instead supplies the frozen teacher-derived contract to every arm. Floating-point solver output is only a proposal. An ACCEPT requires a rational witness that satisfies the original constraints, and a REJECT requires an exact lower certificate, such as a Farkas certificate or a rational structural bound.

Why use recorded timestamps? Each retained retail invoice or taxi pickup contributes one arrival to a recorded stream. Integrating its cumulative count gives

$$p(t) = a + bt + \sum_q z_q (t - t_q)_+,$$

a convex piecewise-linear signal that obeys the sampled dynamics of Section 3. Several arrivals within one bin act as a single effective jump that preserves every sampled position and boundary velocity (Appendix I). We then keep some sample values exact and release others as fixed quantization

Table 6: Evidence map. Each row is a separate protocol with its own denominator; rows are not pooled.

Study	Denominator	Question	Main limitation
D, E	6 and 12 designed inputs; 24 and 48 method–query cells	Q1: can Local DP settle budgets left open by bounds or capped enumeration?	Designed sequentially; different limits
Retail-I	8 windows; 32 inputs; 224 queries per method	Q1 on log-derived shapes	Constructed observation regimes
Retail-II	24 new windows; 96 inputs; 672 queries per method	Q1 on dates fixed before inference	One source; budgets share windows
Taxi-I, Taxi-II	24 cases each; 4 methods $\times$ 3 repeats	Q2: same answers at what complete cost?	Small cases; correlated repeats
Sensitivity	80 cells; 320 attempts	Which anchor and timing changes preserve coverage?	$\beta^+ = 8$ fixed
Core	6 input pairs; 12 attempts	Q3: are local transfer LPs smaller?	3 pairs unresolved
Length	16 designed inputs; 64 cells	Q3: does the pipeline finish as n grows at fixed β?	Larger inputs unresolved
Packing	2 retained inputs; 14 attempts	Q3: does optional packing prevent the continuation from running?	Posthoc selected inputs
Surrogate	12 contracts; 48 method–case records	Do compatible sparse models improve storage at similar predictive error?	One exploratory split; Local DP falls back

cells. The timestamps are recorded data. The loss of precision and the budget task are constructed. An exact checkpoint is exact relative to the digital log, not a claim of noiseless sensing.

What a successful answer means. In Figure 1, a verified two-label witness accepts budget two. This is a compact compatible explanation. It neither recovers the four invoices nor shows that only two arrivals occurred. The theoretical counterexample illustrates the complementary use: a three-event witness together with a lower proof rejects budget two. These budget decisions do not establish downstream utility. The separate surrogate study below tests one possible consumer and delineates a remaining limitation.

G.2 Designed inputs: bounds versus compatible trajectories

Studies D and E use two-interval uncertain clusters separated by s exact regular intervals, with $s \in \{1, 2\}$. Each cluster has a generating total jump of $3/2$ under a unit cap, and only its middle observation is uncertain, so $\beta = 2$. With $s = 2$, exact affine triples fix the boundary velocities and block-cap rejects the supplied budget. With $s = 1$, a compatible explanation can use fewer positive labels than the generator, and only a feasible witness can show that the budget suffices. Appendix H gives all parameters and the reference computation. D has six inputs. E fixes twelve follow-up inputs before execution after D was inspected, so it is not a sample from an independent population. Local DP and enumeration have equal limits within each study, and the bounds-only methods have smaller stated allowances. The comparison therefore measures decision capability under these limits, not speed.

Local DP decides every D and E query, including all eleven feasible budgets. Block-cap decides the seven rejections. Capped enumeration resolves four of six D inputs and six of twelve E inputs and leaves the rest UNKNOWN. All eighteen reference minima were checked independently, and every outcome appears in Figure 4.

Figure 5 shows one accepting certificate. On D05 the Local DP witness uses five events, the generator six, and the exact minimum is five. We chose this example after inspecting D to illustrate the geometry. It is not an additional test or an estimate of typical improvement.

Table 7: ACCEPT / REJECT / UNKNOWN counts on designed inputs, one budget query per input. D and E have exact reference minima. The decisive Length answers have checked budget certificates, though not every minimum is known. All queries are shown.

Method	D: 6 queries	E: 12 queries	Length: 16 queries
Bounds	0 / 0 / 6	0 / 0 / 12	0 / 0 / 16
Bounds + block-cap	0 / 3 / 3	0 / 4 / 8	0 / 4 / 12
Bounds + Local DP	**3 / 3 / 0**	**8 / 4 / 0**	**4 / 4 / 8**
Bounds + enumeration	2 / 2 / 2	4 / 2 / 6	2 / 2 / 12

D: six designed queries

Input	s/k	Bounds	+ cap	+ DP	+ enum
D01	1/3	U	U	**A**	A
D02	1/3	U	U	**A**	A
D03	2/3	U	R	**R**	R
D04	2/3	U	R	**R**	R
D05	1/5	U	U	**A**	U
D06	2/5	U	R	**R**	U

A: compatible witness

R: budget excluded

U: computation incomplete

All cases shown; no pooled rate.
Full-search limits matched within study.

E: twelve follow-up queries

Input	s/k	Bounds	+ cap	+ DP	+ enum
E01	1/3	U	U	**A**	A
E02	1/3	U	U	**A**	A
E03	1/3	U	U	**A**	A
E04	1/3	U	U	**A**	A
E05	1/5	U	U	**A**	U
E06	1/5	U	U	**A**	U
E07	1/5	U	U	**A**	U
E08	1/5	U	U	**A**	U
E09	2/3	U	R	**R**	R
E10	2/3	U	R	**R**	R
E11	2/5	U	R	**R**	U
E12	2/5	U	R	**R**	U

Figure 4: All 72 method–query outcomes in D (24) and E (48). Each row is one input, and columns compare methods on the same budget and constraints. A: certified ACCEPT; R: certified REJECT; U: UNKNOWN at the method's limit. s counts regular intervals between clusters. Local DP's additional decisions occur on the feasible $s = 1$ queries. UNKNOWN for enumeration means an incomplete search, not infeasibility.

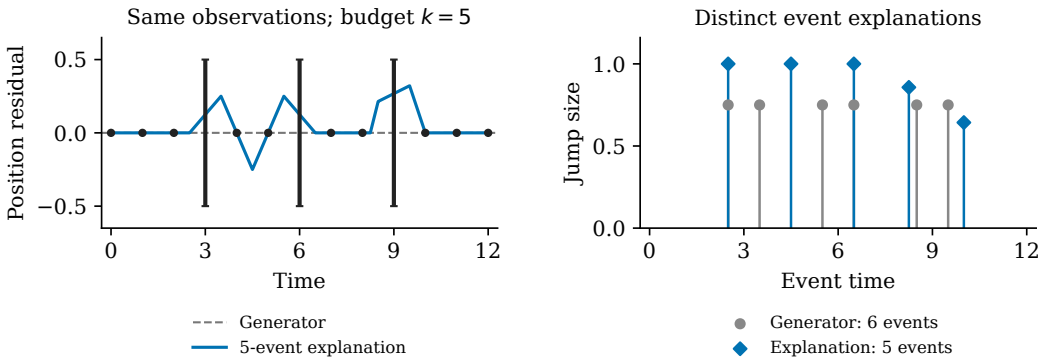

Figure 5: A budget decision on D05, chosen after inspecting all six D inputs. Left: position relative to the generating trajectory, with exact anchors (dots) and three position boxes (bars). Right: positive jump times and sizes. The Local DP witness uses five events, one fewer than the generator, under the same boxes, cap and timing windows; the exact minimum is five. The example illustrates an ACCEPT decision, not latent-event recovery.

G.3 Recorded retail logs: decision coverage and uncertain anchors

Retail-I fixes eight one-hour windows from the UCI Online Retail data (Chen, 2015): the first two eligible 09:00–10:00 dates in each of January, March, June and September 2011. Each hour has six ten-minute labels. Normalizing arrivals by the largest bin count gives $D = J = 1$. Checkpoints 2 and 5 are released with quantization half-width $h \in \{1/8, 1/2\}$, and the other checkpoints with half-width $\epsilon \in \{0, 1/100\}$. The eight windows thus yield four paired regimes and 32 inputs, and every method answers all seven budgets $k = 0, \ldots, 6$ on each: 224 correlated queries per method. The baseline is Bounds + block-cap; the two search methods extend their own bound computation with Local DP or enumeration. Every method receives 30 CPU seconds, 60 wall seconds and 512 solver calls per input, in separate processes with rotating order.

Exact and conditional inference have different guarantees. For $\epsilon = 0$ the instance has $\beta = 2$ and the reset/block algorithm applies directly. For $\epsilon > 0$ every observation is nondegenerate and $\beta = 6$, so Local DP uses conditional anchors: a restricted witness, once checked against the original constraints, is a valid upper certificate, but a restricted lower bound cannot reject the original budget. Lower certificates therefore come only from the original-model bounds. Appendix I proves the conditional-anchor proposition and shows that midpoint pinning can make a feasible problem infeasible.

Local DP resolves 35 of the baseline's 36 UNKNOWN budgets and loses no decision: 223/224 queries, versus 188/224 for the baseline and 224/224 for enumeration. It resolves all 112 exact-anchor queries and 111 of 112 conditional ones. The remaining UNKNOWN is a budget-two rejection that enumeration proves. On all 32 inputs, the upper bound returned by Local DP equals the enumerated minimum; this agreement is retrospective and does not turn the restricted lower bound into an original-model proof. All 96 attempts complete, and their certificates pass rational replay.

New months fixed before inference. Retail-II keeps the Retail-I adapter, quantizer, methods and limits. Before any solver call, it fixes the first six eligible dates in February, April, July and October 2011, none of which supplied Retail-I windows. The 24 new windows yield 96 inputs, 288 attempts and 672 queries per method. Nothing is trained on earlier months; this is a check of the same inference procedure on new inputs.

Local DP adds 126 paired decisions (33, 32, 33 and 28 in the four months), loses none, and leaves six UNKNOWNs that enumeration settles without contradiction (Figure 6). Positive-ϵ cases carry the same conditional-upper limitation as in Retail-I. A zero-solver replay reconstructs all 2,016 method–budget cells.

On these six-label windows, exhaustive enumeration over at most 2^6 supports remains affordable and settles every query. The retail studies therefore test whether the local pipeline is sound and nearly complete on log-derived shapes. They do not test speed, and enumeration's coverage is higher.

G.4 Taxi logs: certified answers and complete cost

The taxi panels address Q2: when certifying methods agree, what does each pay? Taxi-I uses January 2025 NYC Green Taxi pickups (New York City Taxi and Limousine Commission, 2025). A rule fixed before inference selects twelve date/zone/hour windows from January 6–12. Taxi-II applies the same adapter to January 13–19, excludes all previously used window identities, and takes the first twelve of 56 eligible windows in lexicographic order. The result is a same-source extension with disjoint windows, not an independent dataset. Each window gives two cases. The *exact* regime retains all seven ten-minute checkpoints exactly, and the *quantized* regime releases checkpoints 2 and 5 with half-width $1/100$. Every case asks the same budget, $k = 3$, about compatible aggregate labels rather than actual rides. The Taxi-I cases had already been used in a single-repeat measurement and a decision-only run (Appendix I.5), so Taxi-I is an exposed development panel; only Taxi-II uses windows not seen before. Appendix I.5 lists the filtering and every window.

Methods A–C share the Bounds + block-cap front end. Cascade A continues with exact enumeration restricted to supports of size at most three. Cascade B tries Local DP first and falls back to the same enumeration only if needed. Cascade C tries the activation MILP first, accepts only exactly checked witnesses, and sends unresolved cases to the same enumeration. D runs the MILP alone, without

Table 8: Retail-I: ACCEPT / REJECT / UNKNOWN counts. Each row has 56 queries per method from the same eight windows. Local $+/-$ counts decisions added or lost relative to the baseline. Positive ϵ uses conditional anchors.

h	ϵ	Baseline	Local DP	Enumeration	Local $+/-$
1/8	0	25 / 21 / 10	34 / 22 / 0	34 / 22 / 0	10 / 0
1/8	1/100	25 / 21 / 10	34 / 21 / 1	34 / 22 / 0	9 / 0
1/2	0	29 / 19 / 8	37 / 19 / 0	37 / 19 / 0	8 / 0
1/2	1/100	29 / 19 / 8	37 / 19 / 0	37 / 19 / 0	8 / 0

Table 9: Retail-II: outcomes over 672 queries per method, grouped in 24 source windows. Gains and losses are paired against the baseline. Every decisive answer agrees with independently replayed enumeration.

Method	ACCEPT	REJECT	UNKNOWN	Decided	Gain/loss
Bounds + block-cap	311	229	132	540	–
Local DP	431	235	6	666	126 / 0
Enumeration	433	239	0	672	132 / 0

the front end; its positive witnesses are checked, but its negative statuses stay numerical. Each panel runs every case with every method three times, for $24 \times 4 \times 3 = 288$ executions. Each execution has 30 CPU seconds, 60 wall seconds and 1,024 calls; the numerical child is pinned to one CPU with one solver thread. Complete process CPU covers the controller and the reaped child, from payload construction through solving, certificate work and serialization. Eight individually measured controls qualify the instrumentation before scoring, method order rotates deterministically, and every execution stays in the denominator. All 288 records in each panel pass zero-solver verification.

Decisions. Across the 72 executions, each certifying cascade returns 36 ACCEPT and 36 REJECT outcomes on Taxi-I and 45 ACCEPT and 27 REJECT outcomes on Taxi-II, corresponding to 12/12 and 15/9 distinct case outcomes (main-text Table 2). Local DP and the MILP cascade reproduce the enumeration answers throughout. The standalone MILP also returns a status on every case. All its positive answers have checked witnesses, but none of its exclusions is promoted to REJECT. A direct optimizer is therefore a viable alternative on these panels, and the results do not show coverage superiority over it.

Cost. Local DP uses 27.9% and 27.6% less complete CPU than enumeration on the two panels, and never needs the enumeration fallback that A uses 42 and 54 times. Its cost is within 3% of the certifying MILP cascade. The numerical MILP is cheapest, but it offers a weaker guarantee for negative answers. The absolute costs are small, the executions shared a host, and repeats and paired regimes are correlated, so the 288 records per panel are not independent observations. We report the panels separately and infer no population speed ratio. An earlier single-repeat A/B measurement on the Taxi-I cases shows the same direction (7.85 versus 11.42 CPU seconds, with 0 versus 14 fallbacks), and an earlier decision-only run of all four methods gives the same answers. Both appear in Appendix I.5.

A failed reconstruction check. Certified budget answers are not reconstructions. In the single-repeat Taxi-I measurement, the six five-minute midpoint positions were withheld from cascades A and B. For each of the twelve accepted cases, the maximum error over these midpoints exceeds $1/100$ in both cascades, and none of the 24 accepted pipeline rows yields an event encoding smaller than the raw 13-point source series. The witnesses therefore do not serve as a reconstruction or compression of the source signal. This does not affect the certified answers at the released observation points.

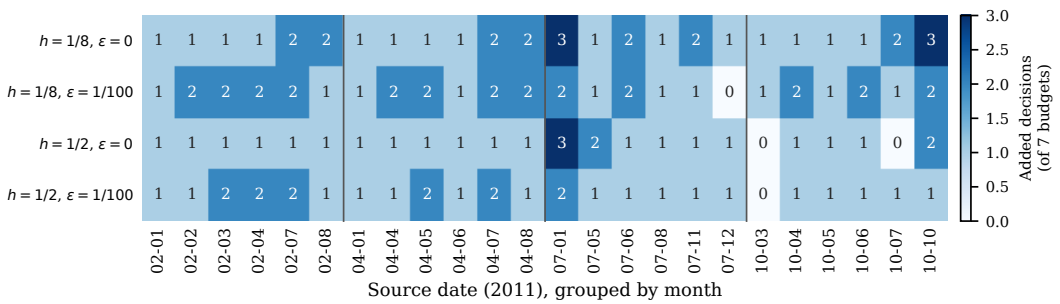

Figure 6: Retail-II: decisions added by Local DP over the paired baseline, for every one of the 96 window–regime inputs (seven budget queries each). Vertical lines separate the four months. No input has a loss. The cells share source windows and are not independent samples.

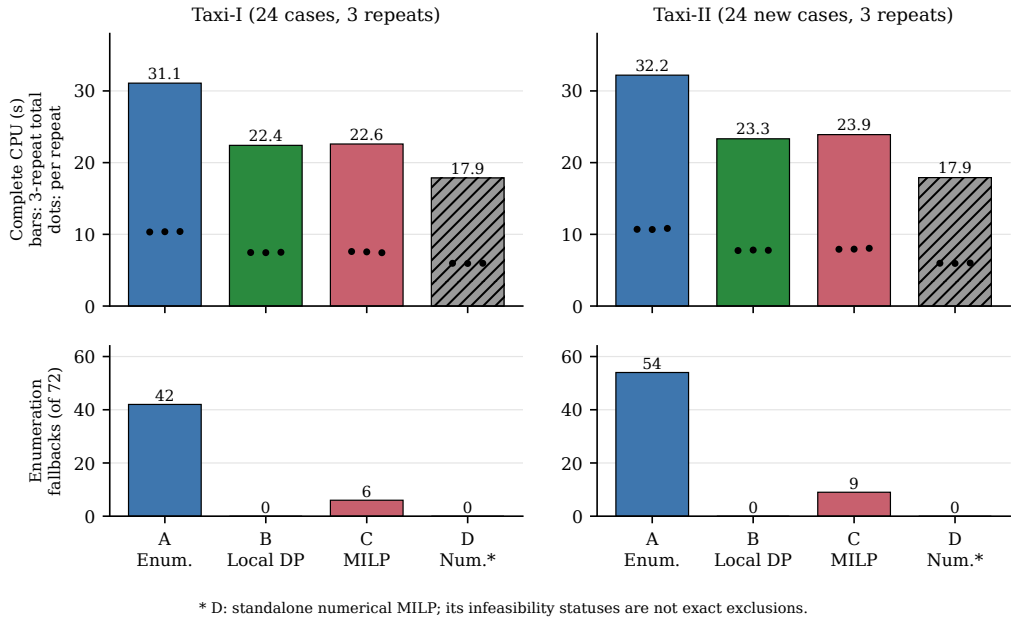

Figure 7: Complete cost on the two taxi panels. Top: CPU totals over three repeats (bars) and per-repeat totals (dots). Bottom: enumeration fallbacks. A–C return the same exact decisions within each panel. Local DP (B) never falls back to enumeration and uses less CPU than enumeration (A): 22.41 versus 31.09 seconds on Taxi-I and 23.33 versus 32.20 on Taxi-II. The MILP cascade (C) costs about the same as B. The standalone numerical MILP (D) is cheaper, but its infeasibility outputs are not exact exclusions. Repeats are correlated and the host was shared, so this is not a general speed benchmark.

G.5 Sensitivity to anchors and timing restrictions

Which conclusions survive changes to the uncertainty contract? Eight fixed rational trajectories on eight labelled intervals vary the generating support over $\{2, 3, 4, 5\}$ and the jump amplitude over $\{1/4, 3/4\}$. Three panels vary the anchor half-width over $\{0, 0.001, 0.01, 0.05\}$, the anchor spacing over $\{2, 4, 8\}$, and the lower timing endpoint over $\{0, 0.25, 0.5\}$. This gives 80 cells (10 settings $\times$ 8 profiles), each queried at budget three by enumeration, Bounds + block-cap, joint-block Local DP and conditional anchors, for 320 attempts. The baseline setting recurs in each panel; those cells are duplicates, not independent trajectories.

Enumeration settles all 80 cells (59 ACCEPT, 21 REJECT), and Bounds + block-cap settles 24 (4 ACCEPT, 20 REJECT). Local DP matches enumeration on all 64 cells that the present implementation supports (48 ACCEPT, 16 REJECT). The other 16 cells, with positive lower timing endpoints,

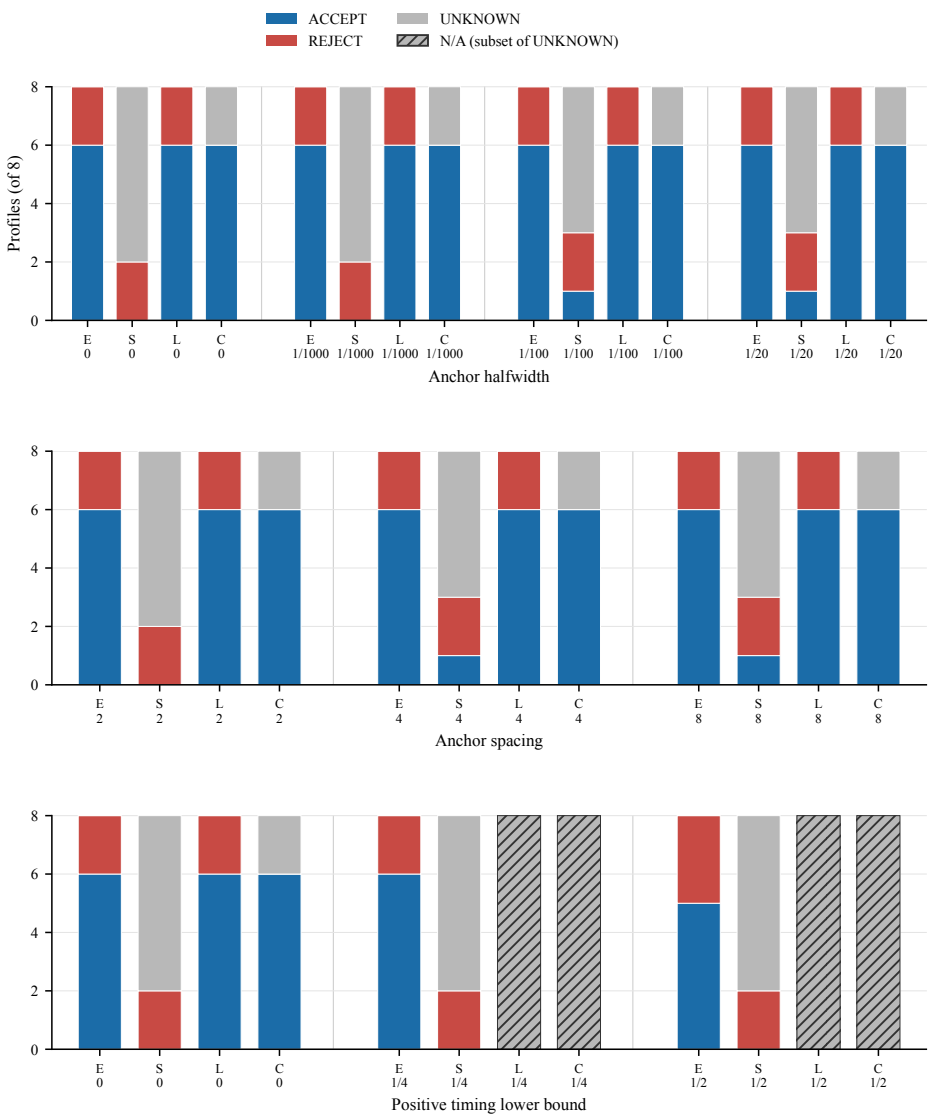

Figure 8: Sensitivity outcomes for eight fixed rational profiles at each setting. Within a setting, the bars are E (enumeration), S (Bounds + block-cap), L (joint-block Local DP) and C (conditional-anchor upper bounds). Colors show ACCEPT, REJECT and UNKNOWN; hatching marks NOT APPLICABLE cells within UNKNOWN, which occur where the local implementation does not support positive lower timing endpoints. Every cell has $\beta^+ = 8$. Repeated baseline cells share the same profiles and are correlated.

are outside the implemented interface; they are recorded as NOT APPLICABLE and counted as UNKNOWN. This is an implementation boundary, not a counterexample to Corollary 9. Conditional anchors yield 48 original-checked witnesses and 32 UNKNOWNs, 16 of which are the same NOT APPLICABLE cells, and never transfer a restricted negative conclusion. In every cell the alternating uncertain positions join all intervals into one block, so $\beta^+ = 8 = n$. The study is therefore a finite coverage check. It does not test scaling in β^+ or tractability with noisy anchors. Appendix Table 19 gives the counts by setting.

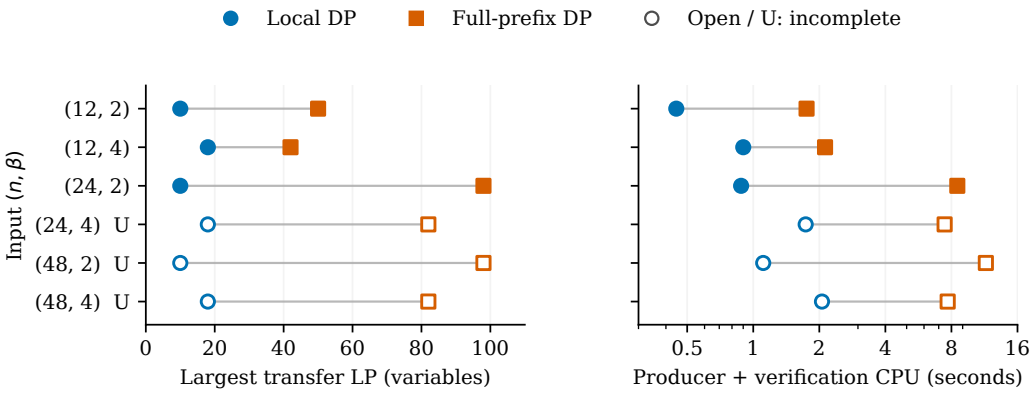

Figure 9: Core panel: six matched pairs. Left: largest transfer LP encountered. Right: CPU including production and verification. Filled markers are exact completions; open markers are UNKNOWN, whose CPU is partial work rather than time to a solution. Local LPs are smaller, but only three pairs complete; this is not an end-to-end speedup estimate.

G.6 Mechanism and scale: what the implementation does and does not show

Smaller transfer problems. Local transfers are meant to retain exact information in smaller optimization problems. An earlier matched panel (Pool C; Appendix J) separates this mechanism from end-to-end performance. Its 24 inputs vary $n \in \{12, 24, 48\}$, $\beta \in \{2, 4\}$, two half-widths and two event placements. All three complete pipelines solve the same 21 inputs in their common front end, and the other three hit the hard cap. Neither local nor full-prefix DP adds a closure, so this main comparison is null. A separate core panel fixed in advance reuses six of these inputs, gives both DP cores the same checked upper witness and budget four, and removes the Bounds front end from both. Three pairs finish, and three are UNKNOWN for both cores. In the finished pairs the largest local LPs have 10, 18 and 10 variables, compared with 50, 42 and 98 for full-prefix projection (Figure 9). This supports a conditional mechanism observation. The core panel does not measure a complete pipeline, and it does not isolate reset merging from other implementation choices.

Length at fixed block size. The Length study increases the number of uncertain clusters b through $2, 4, 8, 16$ with $\beta = 2$ fixed. With two spacings and two amplitudes, its 16 inputs (S1-01 to S1-16) span $n = 9$ to 66. Every method has the same whole-method limits: 45 CPU seconds, 90 wall seconds and 512 calls. Local DP resolves all eight inputs with $b \leq 4$; block-cap and enumeration each resolve four. All methods are UNKNOWN on all eight larger inputs (Figure 10). This is an operating boundary of the current pipeline. The theorem does not predict it, and we do not extrapolate the theorem into an empirical large-scale result.

For one preselected input, a diagnostic trace points to the shared front end rather than the DP. The planned ten-second front-end sublimit was not enforced separately, so only the whole-method limit applied. The 32 completed attempt records were fully replayed, whereas the 32 capped UNKNOWN records received identity and accounting checks only, and where they terminated was not recorded. Four later instrumented traces of one preselected input (S1-09, $n = 27$) find all four methods inside the shared all-triples packing stage when they exhaust the 45-second CPU allowance, before Local DP or the relaxation starts. Those traces used a 45-second rather than 90-second wall cap and identify the active stage for one input only, not the cause of every failure. Three initial attempts on a separately selected Pool C input failed on a missing numerical dependency and supplied no stage attribution. A prospectively fixed recovery in a working environment, with the outer wall allowance changed to 90 seconds, records all three arms completing numerical proposal and certificate reconstruction before entering all-triples packing and reaching that wall limit. No continuation begins. These are three traces of one selected input, not three independent problem instances or a like-for-like timing comparison. The Length study therefore does not measure the scaling of the local algorithm on its own. The following matched ablation isolates one optional preprocessing component on retained inputs.

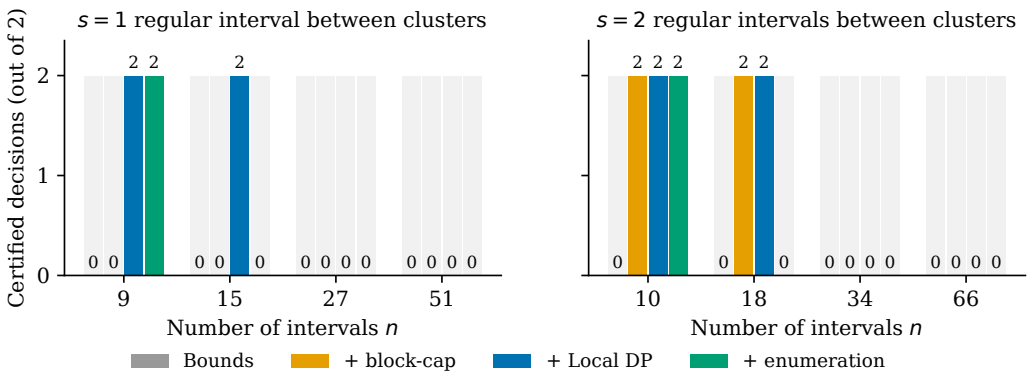

Figure 10: Length study under equal whole-method limits and fixed $\beta = 2$: certified decisions out of two amplitudes at each length. Gray remainders are UNKNOWN. Local DP resolves all inputs with $b \leq 4$; every method is UNKNOWN for $b \in \{8, 16\}$. These runs do not establish successful scaling.

G.7 Can optional preprocessing hide useful local inference?

The original Length failure motivates a controlled mechanism question: can a cheap valid lower bound allow the continuation to reach an answer within the same limit? We use two retained inputs, Length S1-09 ($n = 27$, supplied $k = 15$) and one Pool C input ($n = 48$). They were selected after earlier failures; this is a posthoc diagnostic, not a fresh efficacy panel. Four continuations on the first input and three on the second are each run with all-triples packing ON and OFF, giving 14 attempts. Every attempt has 45 CPU seconds, 90 wall seconds and 512 calls, with one numerical thread. OFF replaces only the optional packing lower bound with the typed bound $K^\star \geq 0$, justified by nonnegative support size. It is not a packing certificate and cannot itself reject a positive budget. The original-model relaxation and witness checks remain in place.

All seven ON attempts reach the CPU cap in all-triples packing before the continuation runs. With packing OFF, Local DP accepts the first input at budget 15 and closes the second at minimum four; the remaining five OFF attempts are UNKNOWN. Both decisive answers pass original-model rational replay. On the first input the retained witness uses 12 positive labels. On the second, the checked four-label upper witness and the local-DP exclusion of smaller supports establish the optimum; its root relaxation alone gives only an integer lower bound of two. Figure 12 shows every attempt, and Appendix L gives the matched table and verification boundaries.

The ablation demonstrates an avoidable obstruction on these two inputs: spending the allowance on a stronger optional bound can prevent an otherwise useful continuation from running. It does not establish a general speed ratio, a scaling law, or superiority on a population of problems. The bars measure child-process CPU under this diagnostic protocol, whereas the taxi panels measure qualified complete-process CPU; their timings are not pooled. Selecting a preprocessing policy before a new structured panel is a separate experiment.

G.8 An exploratory downstream consumer: sparse regression surrogates

A possible consumer of a minimum-event explanation is a compact convex piecewise-linear surrogate. We test this on UCI Auto MPG (Quinlan, 1993), using vehicle weight to predict fuel consumption in gallons per 100 miles. A fixed convex teacher is competitive with simple one-feature predictors on validation data. We then freeze 12 rational observation contracts, crossing 24/48 labels, three approximation tolerances, and retained/released calibration anchors. The dense teacher, a greedy sparsifier, a witness-checked activation MILP, and the Local DP adapter receive the same contract; all candidate identities are frozen before a single opening of 77 held-out targets. Appendix N reports all 48 method–case records, the split, checks and representation correction.

This task admits useful sparse candidates, but the current Local DP adapter does not produce one within its limits: all 12 outputs are UNKNOWN and return the checked dense teacher. MILP reduces canonical storage within the predeclared $1.05\times$ dense-teacher test-MAE limit in 10/12 cases; greedy does so in 6/12. These gains belong to those baselines. Fewer events do not always reduce rational serialization size: two MILP candidates are larger than the dense representation. Prediction timings are too small and variable to support a speed claim. Thus the pilot identifies a plausible consumer of compatible sparse functions while leaving a practical Local DP benefit unresolved.

G.9 Synthesis

The studies support three conclusions. First, exact compatible explanations settle budget queries that cheaper bounds leave open, on designed inputs and on log-derived queries, including 24 retail windows fixed before inference; the conditional-anchor guarantee is weaker and is stated separately. Second, on two taxi-derived panels, Local DP returns the same certified answers as enumeration and a certifying MILP cascade, avoids every enumeration fallback, uses about 28% less complete CPU than enumeration, and costs about the same as the MILP cascade. Third, local LP size and the preprocessing ablation distinguish two implementation issues: a local exact representation can be small while an optional global front end still consumes the budget. The two retained-input successes after removing that stage do not erase the original larger-Length failures. The matched Pool C comparison is null, the taxi reconstruction/storage check fails, and the downstream surrogate pilot finds no Local DP benefit. These boundaries motivate targeted implementation improvements rather than a general deployment claim.

H Constructed budget-query studies

This appendix specifies the complete task family used by D, E and Length. The studies test decision utility within the theorem's model and are not an unbiased performance sample: D was developed after earlier diagnostic work, E was fixed after D was inspected, and Length was fixed after D and E. We keep them separate and report every input in each.

H.1 Inputs and reference minima

Set $D = J = 1$, $g = 0$, initial generating position $p_0 = 0$ and generating velocity $w_0 = 1/8$. The inference problem leaves incoming velocity free. Start with two zero-jump regular intervals, append b two-interval clusters with s zero-jump intervals between consecutive clusters, and end with two regular intervals. Thus $n = 2b + s(b-1) + 4$. Each cluster has generating jump sizes $(a, 3/2 - a)$ at within-interval times $(t, 1-t)$. Core windows are $[1/4, 3/4]$ and regular windows $[0, 1/2]$. Center each observation on the generated sample position. Give only the middle sample of each cluster a box of half-width h; every other box is a singleton. The resulting components have length $\beta = 2$. Every query asks whether $\mathrm{OPT} \le 2b - 1$.

For $s = 1$, the reference solver excludes every smaller support with rational infeasibility certificates and checks a physical witness at the attained minimum. For $s = 2$, exact affine triples and the cap prove a lower bound $2b$, matched by the generator witness. The reference computations are withheld from the competing methods. In particular, the feasible-budget cases cannot be labelled by generator support: their exact minima are $2b - 1$.

H.2 Structural block-cap baseline

Suppose three consecutive exact positions at indices $a, a+1, a+2$ are affine with slope v, the relevant intervals are regular, and $u_{a+1} < D$. Monotone velocity and equality of the two average velocities force $j_{a+1} = 0$ and $w_{a+1} = w_{a+2} = v$. The strict upper-window inequality excludes a terminal jump that would leave the average unchanged. For another eligible triple beginning at c, the difference between these fixed boundary velocities is the sum of nonnegative jumps with labels $a + 2, \dots, c$. Each jump is at most J, so the group requires at least $\lceil \Delta w / J \rceil$ events. Maximizing the sum over disjoint eligible groups gives the implemented block-cap lower bound. On the $s = 2$ family, each cluster changes slope by $3/2$ and therefore needs two unit-capped events. The implementation derives this from observed exact triples, without reading the generating support or reference solution.

Table 10: All fixed inputs and exact reference minima. For D, $a = 3/4$ and $t = 1/2$; for E, $h = 7/16$. The generating support is $2b$ in every row.

ID	b	s	h	a	t	n	k	OPT
D01	2	1	3/8	3/4	1/2	9	3	3
D02	2	1	1/2	3/4	1/2	9	3	3
D03	2	2	3/8	3/4	1/2	10	3	4
D04	2	2	1/2	3/4	1/2	10	3	4
D05	3	1	1/2	3/4	1/2	12	5	5
D06	3	2	1/2	3/4	1/2	14	5	6
E01	2	1	7/16	5/8	1/3	9	3	3
E02	2	1	7/16	5/8	2/3	9	3	3
E03	2	1	7/16	7/8	1/3	9	3	3
E04	2	1	7/16	7/8	2/3	9	3	3
E05	3	1	7/16	5/8	1/3	12	5	5
E06	3	1	7/16	5/8	2/3	12	5	5
E07	3	1	7/16	7/8	1/3	12	5	5
E08	3	1	7/16	7/8	2/3	12	5	5
E09	2	2	7/16	5/8	1/3	10	3	4
E10	2	2	7/16	7/8	1/3	10	3	4
E11	3	2	7/16	5/8	1/3	14	5	6
E12	3	2	7/16	7/8	1/3	14	5	6

H.3 Method outcomes and fairness

Figure 4 contains every method–query pair. No method receives an externally supplied upper witness: the input is a budget query, not a precomputed optimum certificate. Every method pays for its own Bounds computation, and caches and witnesses are not shared across methods. Local DP and enumeration have equal total CPU and call limits within each study, and the bounds-only methods have smaller explicit allowances. D gives Bounds and Bounds + block-cap 30 CPU seconds and 64 calls each, and Local DP and enumeration 120 CPU seconds and 1,024 calls each, including the common allowance. E uses 15 seconds and 64 calls, and 90 seconds and 1,024 calls, respectively, and repeats the four cyclic method orders three times. D and E have different limits and are not pooled.

The four D decisive enumeration queries are D01–D04; D05–D06 are UNKNOWN. The six E decisive enumeration queries are E01–E04 and E09–E10; E05–E08 and E11–E12 are UNKNOWN. Block-cap rejects exactly the $s = 2$ inputs. Local DP returns the reference decision on all inputs. No UNKNOWN is treated as an incorrect certificate, infeasibility result, or completed runtime.

H.4 Evidence verification and operational boundaries

D uses a fixed six-input manifest and 24 attempt records, and E a fixed twelve-input manifest and 48 attempt records. Rational replay checks the saved reference proofs, physical witnesses, structural lower certificates, relaxation evidence, and Local DP or enumeration evidence as applicable. Record identities and source hashes are bound to each record.

E's first independent replay exceeded its time allowance and was interrupted. A separately specified zero-solver replay then checked the unchanged evidence: all 48 attempts, 12 references and two controls passed, as did source and hash checks and a corrupted-dual test; the dropped-record and changed-budget tests at that stage were in-memory smoke checks. The interruption and all recovery costs are recorded in the artifact provenance, and no reported runtime includes this verification work.

H.5 A miniature explanation audit

Consider the member $\lambda = 1/2$ of the family in Theorem 2, whose exact anchors are stipulated by construction. Its supplied observations are

$$(p_0, \ldots, p_5) = \left(\frac{21}{8},\ \left[\frac{13}{8}, \frac{15}{8}\right],\ \frac{7}{8},\ \frac{1}{4},\ 0,\ 0 \right),$$

with $D = J = 1$ and midpoint event windows. Suppose someone proposes that two events explain these observations. A three-event witness starts at $w_0 = -7/8$ and uses jumps $(0, 0, 1/2, 1/4, 1/4)$; it proves feasibility at budget three but does not by itself reject the proposal. The lower argument in Theorem 2 excludes every two-event explanation. Together, the two certificates settle the decision: reject budget two and accept budget three. The relaxed budget-two answer of $\mathcal{F}_2$ is a false explanation. If every budget were excluded, the correct diagnosis would be an inconsistency between model and observations. This worked instance illustrates the output of inference; it is not an application benchmark or a claim about recovering a unique generating history.

H.6 Length: equal whole-method limits at fixed block size

The Length study uses the same generator with $b \in \{2, 4, 8, 16\}$, $s \in \{1, 2\}$ and $a \in \{11/16, 13/16\}$; set $t = 2/5$ and $h = 15/32$. Sort inputs by (b, s, a) and rotate method order as ABCD, BCDA, CDAB, DABC, repeated four times. The resulting 16 inputs span $n = 9$ to 66, all with $\beta = 2$. The full denominator is 64 method–query pairs. No input or limit was changed after results were inspected.

All methods have a hard 45-CPU-second child limit, a 90-second wall watchdog and at most 512 solver calls, including the common front end. The planned common allowance was 10 CPU seconds and 64 calls. In the executed source, packing precedes the relaxation's time check and has no separate 10-second interrupt, so only the whole-method limit bounds that stage, and the planned substage budget was not enforced. The completed small-input common stages used less than one CPU second. The larger failed attempts have no completed common-stage certificate, so where they terminated is not inferred. This does not give any method extra budget, but it prevents using the study as evidence about the local DP independently of its common front end.

Table 11: All 16 Length inputs and all 64 method decisions. $t = 2/5$, $h = 15/32$, $\beta = 2$. A/R/U denote certified ACCEPT, certified REJECT, and UNKNOWN. Every method has 45 CPU seconds and 512 solver calls.

ID	b	s	a	n	k	Bounds	+ cap	+ DP	+ enum
S1-01	2	1	11/16	9	3	U	U	A	A
S1-02	2	1	13/16	9	3	U	U	A	A
S1-03	2	2	11/16	10	3	U	R	R	R
S1-04	2	2	13/16	10	3	U	R	R	R
S1-05	4	1	11/16	15	7	U	U	A	U
S1-06	4	1	13/16	15	7	U	U	A	U
S1-07	4	2	11/16	18	7	U	R	R	U
S1-08	4	2	13/16	18	7	U	R	R	U
S1-09	8	1	11/16	27	15	U	U	U	U
S1-10	8	1	13/16	27	15	U	U	U	U
S1-11	8	2	11/16	34	15	U	U	U	U
S1-12	8	2	13/16	34	15	U	U	U	U
S1-13	16	1	11/16	51	31	U	U	U	U
S1-14	16	1	13/16	51	31	U	U	U	U
S1-15	16	2	11/16	66	31	U	U	U	U
S1-16	16	2	13/16	66	31	U	U	U	U

All four methods are UNKNOWN on inputs S1-09 to S1-16. For S1-01–S1-08, DP has four ACCEPTs and four REJECTs, block-cap has four REJECTs, and enumeration has two ACCEPTs and two REJECTs. Enumeration's remaining four completed attempts exhaust their call limit. Reference generation does not substitute the generating support for the optimum: eight $s = 2$ inputs have analytic budget-rejection references, while the other eight reference entries remain UNKNOWN. Physical witnesses independently certify the six method ACCEPTs without a full optimum computation.

The first verifier was stopped after 32 completed rows because it began recomputing packing on a capped row that had no certificate. A versioned zero-solver correction preserved all scientific outputs and separated full replay of the 32 completed attempts (16 decisive and 16 UNKNOWN) from identity and accounting checks of the 32 capped UNKNOWN attempts; a separate check confirmed this coverage. Copy-only mutations were rejected for one dropped record and one changed budget.

No method was rerun. The compact artifact checks every decisive answer with a physical witness or an analytic block-cap certificate on the same input, which is weaker than replaying every internal DP state or proving every minimum.

The study used approximately 1,608 CPU seconds, or about 2,088 including verification, mutation, export and finalization, within a 3,600-second allowance. Conservative accounting charges 19,499 solver calls, counting each hard-failed attempt at its full 512-call allowance. No GPU or paid API was used. These are evidence-preparation costs, not times to solution.

I DIGITAL TELEMETRY AND CONDITIONAL ANCHORS

I.1 FROM A RECORDED EVENT STREAM TO THE OBSERVATION MODEL

Consider event times t_q and nonnegative weights z_q, recorded at rational precision. Their integrated counting signal is

$$p(t) = a + bt + \sum_q z_q (t - t_q)_+.$$

The affine term accounts for a baseline and arrivals before the displayed window. For labels i of duration D, assign arrivals in $[iD, (i+1)D)$ to label i, and set

$$j_i = \sum_{q: t_q \in [iD,(i+1)D)} z_q, \qquad r_i = \sum_{q: t_q \in [iD,(i+1)D)} z_q\big((i+1)D - t_q\big).$$

With incoming velocity $w_i = b + \sum_{t_q < iD} z_q$, direct subtraction gives $p_{i+1} - p_i = Dw_i + r_i$ and $w_{i+1} - w_i = j_i$. For $j_i > 0$, the effective delay $\tau_i = D - r_i/j_i$ is the weighted mean of the recorded delays in the bin and belongs to $[0, D]$. Thus several arrivals have a single aggregate jump that preserves positions and velocities at every sample. The cap must bound each aggregate j_i. Support counts occupied labels, not individual arrivals.

A value computed and stored from such a log can be an exact rational checkpoint relative to that log. This interpretation does not require noiseless physical position sensing, and does not establish that the log records every real event. For low-precision telemetry, an observation box can instead be a quantization cell containing the exact digital value. Its center generally differs from that value.

I.2 WHAT REMAINS SAFE WHEN ANCHORS ARE UNCERTAIN

Proposition 10 (Conditional-anchor certificates)**.** *Let $\mathcal{P}$ be the original feasible set under* (1)–(3)*. Fix selected position coordinates to rational values $c_i \in [a_i, b_i]$, giving a restricted set $\mathcal{P}_c$. Then*

$$\mathrm{OPT}(\mathcal{P}) \le \mathrm{OPT}(\mathcal{P}_c).$$

If a restricted solve returns a physical witness of support U, that witness certifies $\mathrm{OPT}(\mathcal{P}) \le U$. For any sound lower bound L on the original problem, $k \ge U$ is safely accepted and $k < L$ safely rejected. A lower bound or infeasibility result for $\mathcal{P}_c$ alone cannot reject a budget for $\mathcal{P}$.

Proof. The restriction only removes feasible trajectories, so $\mathcal{P}_c \subseteq \mathcal{P}$. Minimization over the larger set cannot increase the objective. The two decisions follow from $L \le \mathrm{OPT}(\mathcal{P}) \le U$. □

The distinction is necessary even for midpoint anchors. Let $D = J = 1$, both windows be $[0, 1]$, and the three position boxes be $[0, 0], [0, 2], [0, 2]$. The zero trajectory is feasible with no jumps. Fixing all positions to the box centers instead forces $(0, 1, 1)$, whose chord slopes decrease, contrary to the nonnegative-jump model. Thus the midpoint-restricted problem is infeasible although the original optimum is zero. A failed conditional-anchor solve returns UNKNOWN unless an original-model lower certificate independently settles the query.

If the selected anchors produce small reset blocks, the restricted solve can use Theorem 6. This provides a route to feasible sparse explanations for nondegenerate boxes; it is not an exact small-block guarantee for the unrestricted noisy-anchor problem. Every returned witness must be checked in the original boxes.

I.3 RETAIL-I: COMPLETE STUDY

We use the UCI Online Retail dataset (Chen, 2015), distributed under CC BY 4.0. Each retained unit is one deduplicated, non-cancelled invoice with at least one positive-quantity item row. We exclude units whose retained rows disagree on timestamp. Of 541,909 item rows, the adapter identifies 20,728 candidate invoice IDs and excludes 42 with conflicting timestamps, retaining 20,686 unit arrivals. The experiment exports timestamps and aggregate counts, without invoice IDs, customer IDs, product descriptions or original item rows. Times retain the workbook's recorded local civil convention; we perform no timezone conversion or physical-arrival imputation.

Before outcomes, we fixed January, March, June and September 2011 and selected the first two dates in each month with at least one retained invoice in $[09{:}00, 10{:}00)$. No input was replaced by difficulty or result. The resulting source windows are below; each contains six half-open ten-minute bins. We set $D = J = 1$, divide all unit weights by the window's largest bin count M, and construct $p(t)$ as above with zero generating baseline. Inference leaves initial velocity free. Thus $J = 1$ is a data-defined normalized cap, not an assumed operational transaction-rate bound.

Window	Date	Invoices	Occupied bins G
W01	2011-01-05	4	3
W02	2011-01-06	3	3
W03	2011-03-01	7	5
W04	2011-03-02	6	4
W05	2011-06-01	3	3
W06	2011-06-02	8	3
W07	2011-09-01	5	4
W08	2011-09-02	10	5

Samples $i \in \{2, 5\}$ have half-width $h \in \{1/8, 1/2\}$; others have $\epsilon \in \{0, 1/100\}$. For a positive half-width a, the released cell is $[c - a, c + a]$ with $c = 2a\lfloor p_i/(2a) + 1/2 \rfloor$, using exact rational half-up quantization. At zero width it retains the digital value. The source positions and generating witness are withheld from every method. All timing windows are $[0, 1]$. The original problem has $\beta = 2$ for $\epsilon = 0$ and $\beta = 6$ for positive ϵ. In the latter regime, the conditional method pins the five sample positions outside $\{2, 5\}$ to their supplied centers. No hidden source value is used for that restriction.

Methods run in separate processes and rotate as ABC/BCA/CAB across the fixed input order. The equal per-attempt limits are 30 CPU seconds, 60 wall seconds, 512 solver calls, one thread and 4 GiB; there is no separate common-stage time claim. Every method may use a physical upper witness already present in its own relaxation solution: activation-only variables are projected out and the original constraints are checked exactly. Enumeration covers every support size from zero to six and provides the reference only after all necessary exclusions and the attaining witness have been replayed. References are not supplied to other methods. The two-window startup subset belongs to the same fixed 32-input denominator, and nothing was tuned on its outcomes.

All 96 attempts completed, using 3,034 solver calls, approximately 31.51 CPU seconds and 52.87 wall seconds. A separate zero-solver replay checked all 96 records, with its allowance charged conservatively at 60 CPU seconds. Independent checks recomputed the log integration, quantization, physical-witness constraints, record multiplicities and all 672 displayed decisions; they complement, and do not replace, the replay of lower bounds, DP states and exclusions. Source, input, code, raw-certificate and result hashes bind the artifacts. Adapter repairs made before inputs were fixed are recorded separately, and no scientific attempt was rerun or filtered.

The table below contains all inputs. Here m_C is the original-model minimum established by enumeration and u_B the best original-checked upper bound returned by Local DP, including its own relaxation witness. A/R/U count that method's seven budget decisions. A dash would denote an unavailable certificate. Every returned u_B equals m_C on this panel; this is an external reference comparison, not a claim that Local DP proved every original optimum. The single UNKNOWN is W01 with $h = 1/8$, $\epsilon = 1/100$ and $k = 2$: its original lower bound is two and upper is three, while enumeration proves the minimum is three. The conditional lower restriction is therefore material. Across the 32 paired instances, 26 minima are below G and six equal G; none of these counts estimates latent invoice recovery or population performance.

Window	h	ϵ	M	G	Bin counts	m_C	u_B	A	R	U
W01	1/8	0	2	3	0,1,0,0,1,2	3	3	4	3	0
W01	1/8	1/100	2	3	0,1,0,0,1,2	3	3	4	2	1
W01	1/2	0	2	3	0,1,0,0,1,2	2	2	5	2	0
W01	1/2	1/100	2	3	0,1,0,0,1,2	2	2	5	2	0
W02	1/8	0	1	3	1,0,0,0,1,1	2	2	5	2	0
W02	1/8	1/100	1	3	1,0,0,0,1,1	2	2	5	2	0
W02	1/2	0	1	3	1,0,0,0,1,1	2	2	5	2	0
W02	1/2	1/100	1	3	1,0,0,0,1,1	2	2	5	2	0
W03	1/8	0	2	5	1,0,1,2,1,2	3	3	4	3	0
W03	1/8	1/100	2	5	1,0,1,2,1,2	3	3	4	3	0
W03	1/2	0	2	5	1,0,1,2,1,2	3	3	4	3	0
W03	1/2	1/100	2	5	1,0,1,2,1,2	3	3	4	3	0
W04	1/8	0	3	4	3,0,1,1,1,0	3	3	4	3	0
W04	1/8	1/100	3	4	3,0,1,1,1,0	3	3	4	3	0
W04	1/2	0	3	4	3,0,1,1,1,0	2	2	5	2	0
W04	1/2	1/100	3	4	3,0,1,1,1,0	2	2	5	2	0
W05	1/8	0	1	3	1,1,1,0,0,0	3	3	4	3	0
W05	1/8	1/100	1	3	1,1,1,0,0,0	3	3	4	3	0
W05	1/2	0	1	3	1,1,1,0,0,0	3	3	4	3	0
W05	1/2	1/100	1	3	1,1,1,0,0,0	3	3	4	3	0
W06	1/8	0	4	3	4,0,0,2,2,0	2	2	5	2	0
W06	1/8	1/100	4	3	4,0,0,2,2,0	2	2	5	2	0
W06	1/2	0	4	3	4,0,0,2,2,0	2	2	5	2	0
W06	1/2	1/100	4	3	4,0,0,2,2,0	2	2	5	2	0
W07	1/8	0	2	4	1,2,0,1,0,1	2	2	5	2	0
W07	1/8	1/100	2	4	1,2,0,1,0,1	2	2	5	2	0
W07	1/2	0	2	4	1,2,0,1,0,1	2	2	5	2	0
W07	1/2	1/100	2	4	1,2,0,1,0,1	2	2	5	2	0
W08	1/8	0	3	5	1,0,2,3,3,1	4	4	3	4	0
W08	1/8	1/100	3	5	1,0,2,3,3,1	4	4	3	4	0
W08	1/2	0	3	5	1,0,2,3,3,1	3	3	4	3	0
W08	1/2	1/100	3	5	1,0,2,3,3,1	3	3	4	3	0

I.4 RETAIL-II: NEW MONTHS FIXED BEFORE INFERENCE

Retail-II keeps the adapter, quantizer, methods, limits and certificate rules of Retail-I, but selects the first six eligible $[09{:}00, 10{:}00)$ dates in each of four unused months before any solve. It therefore has 24 source windows and four released regimes per window. All 96 inputs and 288 attempts complete, and the independent zero-solver replay reconstructs all 2,016 budget decisions. Figure 6 shows the decisions added by Local DP over the Bounds + block-cap baseline for every window and regime: 126 in total, with no losses and six remaining Local DP UNKNOWNs. The 96 cells are not independent population observations.

I.5 TAXI PANELS: DATA, SINGLE-REPEAT MEASUREMENT AND DECISION-ONLY MILP RUN

The taxi panels use the January 2025 Green Taxi Trip Record file from the official NYC Taxi and Limousine Commission Trip Record Data page (New York City Taxi and Limousine Commission, 2025). The adapter reads only the pickup timestamp and pickup-zone fields. Of 48,326 file rows, the adapter retains 48,160 January rows with published integer pickup-zone IDs 1–263; it excludes 43 rows outside January and 123 invalid zone IDs, and retains all 904 coincident-timestamp rows as distinct source-row ordinals (no timestamp deduplication). The provider-published Parquet is retained remotely under SHA-256 `84f3a121667157efcbf012c3566a6065df6f8e0312c678cb2f29cd72cc9c0f10`; the archived input manifest records its URL, fields, filters, exact rational inputs, and source hash. The data are provider-submitted records under NYC Open Data/TLC terms. They support the constructed observation task, not statements about demand or about actual rides.

Before any decision process, the Taxi-I rule enumerates the date/zone/hour keys of January 6–12, 2025 at hours 08, 12 and 18, retains keys with 8–40 valid pickups, sorts them lexicographically by date, zone and hour, and fixes the first 12 of 53 eligible windows (Table 13). Taxi-II applies the

Table 13: Taxi-I source windows. The selection rule, not outcomes, determines these 12 rows.

ID	Date	Pickup zone	Hour / pickups
W01	2025-01-06	43	18 / 15
W02	2025-01-06	74	12 / 20
W03	2025-01-06	74	18 / 24
W04	2025-01-06	75	08 / 12
W05	2025-01-06	75	12 / 11
W06	2025-01-06	75	18 / 27
W07	2025-01-06	97	18 / 14
W08	2025-01-06	166	12 / 8
W09	2025-01-07	43	18 / 18
W10	2025-01-07	74	12 / 15
W11	2025-01-07	74	18 / 34
W12	2025-01-07	75	08 / 8

same adapter to January 13–19, excludes every window identity used before, and keeps the first 12 of 56 eligible windows in lexicographic order: seven on January 13 and five on January 14. Each selected hour is partitioned into six ten-minute labels. We divide its recorded bin counts by that window's largest bin count M, integrate the resulting rational stream with zero generating baseline, set $D = J = 1$, and leave the initial velocity free. The exact regime releases seven exact ten-minute checkpoints. The quantized regime instead releases half-up centers with half-width $1/100$ at checkpoints 2 and 5 and leaves the others exact. Thus 12 windows and two regimes give 24 cases per panel, each asking whether a compatible trajectory exists with budget $k = 3$. This is a compatible-event-budget task, not an estimate of actual rides, their times, or their count.

Single-repeat measurement. The first taxi experiment ran cascades A and B once on the 24 Taxi-I cases. Cascade A runs the Bounds + block-cap front end and then a budget-aware exact enumeration for supports of size at most three. Cascade B runs the same front end, then Local DP, then the same enumeration if needed. Processes are freshly started, share no cache or reference, and receive only the case identifier, regime, budget $k = 3$ and released observations; source arrivals, generating data and the six withheld five-minute midpoint positions are unavailable to them. Each attempt has 60 CPU seconds, 90 wall seconds, 1,024 solver calls and one thread. The records establish a single-host execution but do not preserve processor model or RAM. Complete process CPU is measured from before payload construction and process creation until after child reaping, verification of the original certificates and evidence serialization, as the controller's CPU delta plus those of the reaped child processes. All 48 attempts complete, with the same 12 ACCEPT and 12 REJECT decisions per cascade (Table 14, Figure 11). B uses 31.3% less CPU, about 3.57 seconds in total. An additional 0.244 seconds of controller work lies outside the per-attempt boundary; splitting it evenly gives a 31.0% difference. This is one repeat on one host, so it cannot establish a stable expected speedup; the repeated panels in Section G.4 are the primary cost evidence, and this record is kept as measured. An earlier child-process-only timing of the same run is not used.

Table 14: Single-repeat complete-process measurement on the 24 Taxi-I cases. CPU includes controller and child work from payload construction through certificate verification and serialization.

Measurement	A: front end, then enumeration	B: front end, then Local DP
ACCEPT / REJECT / UNKNOWN	12 / 12 / 0	12 / 12 / 0
Enumeration fallbacks	14	0
Solver calls	444	296
Complete process CPU (s)	11.42	7.85

Reconstruction and storage check. This check was computed after the decisions and is not an input to either cascade. For all 12 accepted cases, the maximum reconstruction error over the six withheld five-minute midpoints exceeds $1/100$ for each cascade. Across the 24 accepted pipeline rows (two cascades per accepted case), no event encoding is smaller than the raw 13-point half-step

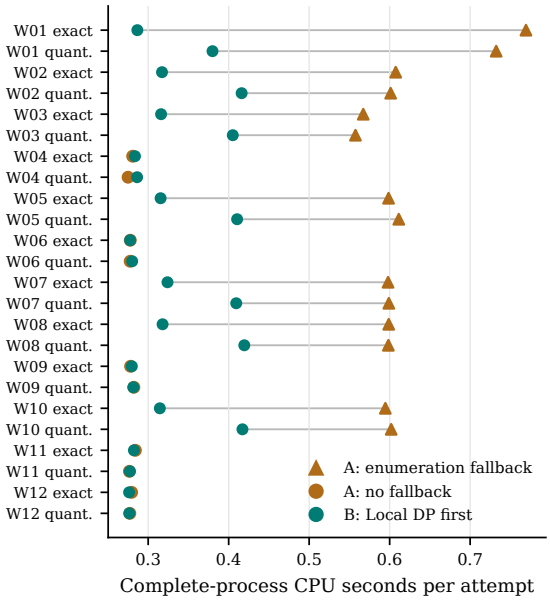

Figure 11: Single-repeat measurement: all 24 Taxi-I case pairs. Lines join matched cases; triangles mark the 14 cases in which cascade A needed the enumeration fallback. One repetition on one host; not a speed benchmark.

source series. The check therefore fails; it does not affect the certified $k = 3$ decisions at the released observations.

Decision-only run of all four methods. Before the repeated panels, all four methods of Section G.4 were run once on the 24 Taxi-I cases, with 30 CPU seconds, 60 wall seconds and 1,024 calls per attempt and eight small encoding controls beforehand. Its decisions are complete and replayed (Table 15). Its controls lack individual CPU, wall and call fields, so its timings are not used. This is a comparison on an exposed development panel, not an independent replication.

Table 15: Decision-only run on the 24 Taxi-I cases. Every positive witness is checked against the original rational model and budget, and every exact exclusion is replayed. The standalone MILP's numerical infeasibility statuses are kept at their own evidence level. No runtime comparison is made.

Method	Verified ACCEPT	Exact REJECT	Numerical negative only	UNKNOWN
A: front end → enumeration	12	12	0	0
B: front end → Local DP → enumeration	12	12	0	0
C: front end → MILP → enumeration	12	12	0	0
D: standalone MILP, positive witnesses checked	12	0	12	0

J EARLIER IMPLEMENTATION DIAGNOSTICS

These studies preceded D, E and the recorded-log panels. They concern earlier formulations or matched implementation controls, and their outcomes are not pooled with later studies. Pool A has twelve inputs and two methods; Pool B has twelve different inputs and three methods; Pool C has twenty-four inputs and three main methods, plus the twelve core attempts on six reused inputs described in Section 6. Their combined inventory is 144 attempts, not 144 independent inputs.

Table 16: Exact closures in the earlier pools. Parentheses count additions beyond the common Bounds front end. The two Pool A additions come from the separator algorithm of Appendix E, not from the reset/block implementation.

Continuation	A: 12 inputs	B: 12 inputs	C: 24 inputs
Enumeration	9 (0)	8 (0)	21 (0)
Local separators	11 (2)	8 (0)	–
Full-prefix DP	–	8 (0)	21 (0)
Local DP	–	–	21 (0)

Pool A permits midpoint regular windows, outside the reset condition. Its two separator completions tighten bounds $[3, 4]$ to the exact minimum four; the larger residual input remains incomplete. In Pool B, full-prefix DP adds no closures. In Pool C, the common front end closes every successful main input, so the main comparison is null for the local implementation. The core panel exhibits smaller local transfer LPs but only three completed pairs. Tables 17 and 18 list every Pool C input and core outcome. No speed estimate treats a capped result as a completed solution.

The Pool C check covers 84 saved attempts: 63 replays of common rational certificates, 12 core outcomes and nine censoring records, with no new solver calls. Earlier checker repairs and the reconstructed startup-call accounting are part of the archived protocol and are not additional scientific runs. This paper reuses these outcomes without rerunning them or strengthening the earlier verification claims.

Table 17: All twenty-four Pool C inputs. The three main methods agree in each row. Optimum U denotes three hard-cap UNKNOWN outcomes. Seeds are 202609211700 plus the row index.

Index	n	β	Width	Placement	Exact methods	Optimum
0	12	2	1/32	uniform	3/3	4
1	12	2	1/32	clustered	3/3	4
2	12	2	1/4	uniform	3/3	3
3	12	2	1/4	clustered	3/3	3
4	12	4	1/32	uniform	3/3	3
5	12	4	1/32	clustered	3/3	3
6	12	4	1/4	uniform	3/3	3
7	12	4	1/4	clustered	3/3	3
8	24	2	1/32	uniform	3/3	4
9	24	2	1/32	clustered	3/3	4
10	24	2	1/4	uniform	3/3	3
11	24	2	1/4	clustered	3/3	4
12	24	4	1/32	uniform	3/3	4
13	24	4	1/32	clustered	3/3	4
14	24	4	1/4	uniform	3/3	4
15	24	4	1/4	clustered	3/3	3
16	48	2	1/32	uniform	0/3	U
17	48	2	1/32	clustered	0/3	U
18	48	2	1/4	uniform	0/3	U
19	48	2	1/4	clustered	3/3	4
20	48	4	1/32	uniform	3/3	4
21	48	4	1/32	clustered	3/3	4
22	48	4	1/4	uniform	3/3	4
23	48	4	1/4	clustered	3/3	4

Table 18: All twelve conditional core attempts. CPU includes production and verification. U denotes UNKNOWN; recorded CPU and visited LP sizes on those rows describe partial work, not a completed solve.

n	β	Core	Vars.	Constr.	Calls	CPU (s)	Status
12	2	local	10	32	87	0.446	exact
12	2	prefix	50	158	97	1.750	exact
12	4	local	18	60	133	0.900	exact
12	4	prefix	42	134	135	2.124	exact
24	2	local	10	32	165	0.880	exact
24	2	prefix	98	322	217	8.509	exact
24	4	local	18	60	256	1.733	U
24	4	prefix	82	274	256	7.449	U
48	2	local	10	32	254	1.110	U
48	2	prefix	98	330	254	11.480	U
48	4	local	18	60	256	2.057	U
48	4	prefix	82	278	256	7.691	U

K FULL SENSITIVITY OUTCOMES

Each setting reuses the eight fixed rational profiles. The joint-block parameter is eight throughout, and baseline cells repeated across panels are not independent evidence. The NOT APPLICABLE cells at positive lower timing endpoints mark the boundary of the current implementation, not a failure of Corollary 9.

Table 19: Sensitivity outcome counts for eight constructed rational profiles per setting. Each entry is ACCEPT/REJECT/UNKNOWN; a parenthesized N/A count is included in UNKNOWN, not an additional denominator. The conditional-anchor method reports only original-checked feasible upper witnesses, so it never rejects.

Panel	Setting	Enumeration	Bounds + cap	Joint DP	Conditional upper
Anchor halfwidth	0	6/2/0	0/2/6	6/2/0	6/0/2
	$1/1000$	6/2/0	0/2/6	6/2/0	6/0/2
	$1/100$	6/2/0	1/2/5	6/2/0	6/0/2
	$1/20$	6/2/0	1/2/5	6/2/0	6/0/2
Anchor spacing	2	6/2/0	0/2/6	6/2/0	6/0/2
	4	6/2/0	1/2/5	6/2/0	6/0/2
	8	6/2/0	1/2/5	6/2/0	6/0/2
Timing lower bound	0	6/2/0	0/2/6	6/2/0	6/0/2
	$1/4$	6/2/0	0/2/6	0/0/8 (8 N/A)	0/0/8 (8 N/A)
	$1/2$	5/3/0	0/2/6	0/0/8 (8 N/A)	0/0/8 (8 N/A)

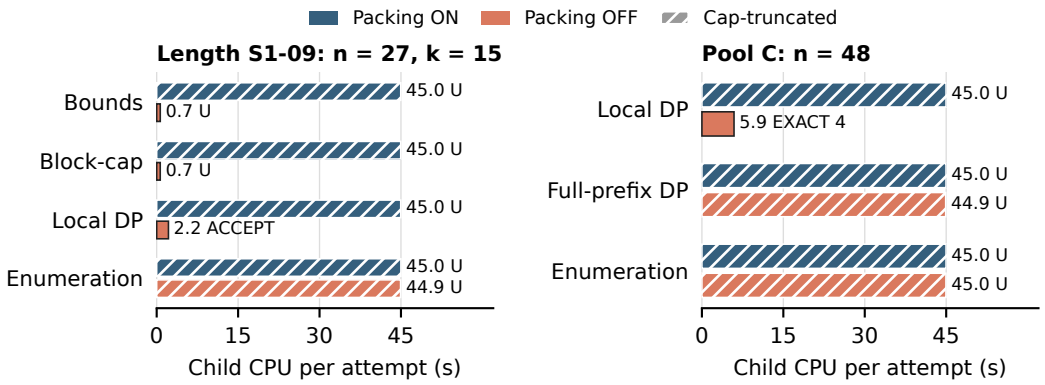

Figure 12: Packing ablation on two retained inputs, selected after earlier failures: child CPU of all 14 matched attempts. Hatched bars hit the 45-second cap (UNKNOWN). With packing OFF, Local DP accepts the Length input and closes the Pool C input at its exact minimum of four. A finite mechanism observation, not a scaling benchmark.

L RETAINED-INPUT PREPROCESSING ABLATION

The preprocessing study uses the same selected Length S1-09 and Pool C observations in both conditions. Selection followed earlier failures; all 14 planned attempts are reported, including capped attempts and missing terminal payloads. Packing ON enumerates all observation triples for its lower bound. Packing OFF supplies a distinct, typed nonnegative-support lower bound of zero with zero packing work. It neither fabricates a packing certificate nor changes the original-model constraints. Six outcome-blind controls checked the zero-bound interface, rejection of fabricated nonzero bounds, tiny known feasible/infeasible cases, inability to reject from missing packing, and watchdog/trace handling.

Table 20: All seven matched ON/OFF pairs, totaling 14 attempts. CPU is child-process CPU in seconds, not the qualified complete-process measure of the taxi panels. U: UNKNOWN, A: ACCEPT, E(4): exact minimum four; superscript cap marks a truncated attempt. A cap is not an exact negative. Calls are charged calls, including conservative cap charges on missing payloads. The Length query supplies $k = 15$; the Pool C Local DP result closes the minimum at four.

Panel	Method	ON CPU / status	OFF CPU / status	ON calls	OFF calls
Length	Bounds	44.962 / U^{cap}	0.661 / U	512	2
Length	Block-cap	44.957 / U^{cap}	0.656 / U	512	2
Length	Local DP	44.957 / U^{cap}	2.168 / A	512	149
Length	Enumeration	44.989 / U^{cap}	44.930 / U^{cap}	512	512
Pool C	Local DP	44.975 / U^{cap}	5.917 / E(4)	512	353
Pool C	Full-prefix DP	44.971 / U^{cap}	44.909 / U^{cap}	512	512
Pool C	Enumeration	44.976 / U^{cap}	44.955 / U^{cap}	512	512

For S1-09, the exact original-model root lower bound is 12 and the retained Local DP witness has 12 positive labels, within the supplied budget 15. Zero-solver replay checks its original constraints and 44 local transitions. For the Pool C input, the root relaxation value is $3/2$, which gives the integer lower bound two; a four-label upper witness alone would therefore not close the problem. Replay of the Local DP trace checks 90 transitions and the exclusion of smaller budgets, and verifies the four-label original-model witness. The checks consume no new solver calls. The other twelve attempts remain UNKNOWN. For missing terminal payloads the call ledger conservatively charges the allowance, rather than claiming an observed call count.

The full diagnostic, including controls, records 468.36 child-CPU seconds, 496.75 parent-wall seconds and 6,078 charged calls. These total categories are different from per-attempt costs and from the taxi instrument's complete-process CPU. The host remained shared. The final table is reconstructed from retained attempt traces without another solve. Incomplete payloads remain UN-

KNOWN; operational failures and their original logs remain in the evidence archive. The scoped conclusion is that the optional packing stage prevented the continuation from running on these two inputs under this allowance. Neither diagnostic selection nor an implementation repair turns these rows into a fresh scaling benchmark.

M LOCAL VERSUS FULL-PREFIX PROJECTIONS

A separate prospective mechanism probe compares the local-transfer and full-prefix DP implementations directly, with no packing, shared bound, enumeration fallback or cross-method certificate reuse. Four fixed queries use $n = 12$, $D = 1$, $g = 0$, $J = 1/4$, an interior uncertainty block of size $\beta \in \{2, 4\}$, and budgets $k \in \{3, 4\}$. The constructed trajectory has four jumps; it supplies a feasibility construction, not a minimum label or an observed initial velocity. The source and inputs were fixed before execution. Methods alternate order across queries and have the same nominal 20 CPU-second, 30 wall-second and 128-call allowances. Eight analytic controls and two zero-call controls precede the panel; every output, including the four bounded enumeration references, is retained.

Table 3 in the main text reports every query.

All eight analytic controls and both zero-call controls pass. A fresh execution of the frozen zero-solver checker replays all 20 ordinary outputs, and certificate, budget, status, input and missing-record mutations are rejected. This is a separate checker execution, not an independently implemented verifier. Local and full-prefix DP settle all four queries and agree; capped enumeration supplies no reference minimum. The local LPs are smaller and use fewer calls in each pair (Table 3), while the peak retained-record counts are identical within pairs: 2, 2, 4 and 7. This distinguishes reduced projection size from a reduction in the observed state count.

The implementations have different recurrence cores, so the comparison does not isolate the causal effect of reset merging or establish a general speedup. The eight paired solver-body CPU times sum to 1.1510 seconds for local DP and 2.3603 seconds for full-prefix DP, but these timers exclude imports, launch and controller work and are not end-to-end costs. All 22 outputs, including references and controls, charge 886 calls; their solver-body CPU totals 20.4301 seconds. Separate zero-solver replay uses 6.9984 CPU seconds. The wrapper enforces per-child CPU limits with one/two-second soft/hard grace and a two-second wall grace; complete whole-program CPU was not metered. We therefore use the dimension and call-count observations as finite mechanism evidence, rather than promoting these timings to the qualified taxi cost comparison.

N EXPLORATORY SPARSE-SURROGATE CONSUMER

Task and suitability. The UCI Auto MPG data supply 398 vehicle records. We use weight as the sole input and $100/\mathrm{mpg}$ as the target, in gallons per 100 miles. No weight or mpg value is missing in the retained source. Name-based approximate vehicle families are assigned as whole groups using seed 2026092301: 239 training rows, 82 validation rows and 77 test rows, with no group overlap. The name grouping is not externally validated, so this is not evidence of clean unseen-family generalization. Two single-token names caused an initial parsing failure; their deterministic group rule was fixed before fitting, with the failed preparation retained.

A convex hinge teacher uses ten training-weight quantile knots and nonnegative hinge coefficients. Affine least squares and a predeclared unconstrained piecewise-linear grid (4/10/16 knots and hinge-ridge penalties 0/0.001/0.01) provide validation context. Train-only normalization is used. The convex teacher's validation MAE is 0.558807, compared with 0.563784 for affine and 0.557209 for the selected unconstrained model. It passes the prespecified suitability rule of no more than 1.10 times the better baseline's validation MAE. This establishes adequacy for the exploratory adapter, not superiority of convexity.

Observation contract and candidates. The frozen teacher induces 24 or 48 equal-width labels on the normalized training input domain. Its grid values are rationalized, and observation half-widths are 1%, 2% or 5% of the training-target IQR. Each tolerance is tested with exact teacher calibration anchors at endpoints and every eighth label, and with the interior anchor equalities released: twelve contracts. The jump cap comes from the largest teacher grid-slope jump. Every dense grid teacher

has eight positive jumps and passes the original rational constraints. Calibration anchors describe a supplied teacher contract; they are not exact observations of the noisy response.

Each contract is processed by the dense teacher, a greedy boundary-event sparsifier, a witness-checked activation MILP and our conditional Local DP adapter. The latter inserts exact conditional positions to obtain short blocks; only a resulting witness checked against the original contract can be accepted. A restricted negative cannot reject the original problem. Each search arm has 120 CPU seconds, 240 wall seconds, 8,192 calls, one numerical thread and 4 GiB per case. Local DP searches $k = 0, \ldots, 8$ with a nominal 12-second wall allowance and 1,000 endpoint calls per budget, within the case envelope. All UNKNOWN and capped outcomes fall back to the checked dense grid teacher, without counting that fallback as a successful sparse construction.

Representation and test boundary. Canonical storage serializes the exact rational function with a common endpoint convention. Prediction uses its numerical realization. These are separate costs, and fewer jumps need not produce fewer serialized bytes. A pretest canonicalization defect used an exterior endpoint derivative instead of the first/last in-domain segment slope. Three synthetic controls established the correction; all 48 frozen primals were deterministically recanonicalized without another solver call. The defect was latent on these primals, which had no endpoint slope changes. The original representation and corrected lineage remain retained.

All candidate identities, exact checks, validation results, timing runs and evaluator code were frozen before opening the 77 held-out targets once. Table 21 retains every case and arm. The threshold for a storage-preserving predictive result was smaller canonical size and test MAE no larger than 1.05 times the corresponding dense grid teacher. This exploratory tolerance is not a significance test. Test results did not select a new model or trigger another test opening.

Table 21: All twelve surrogate contracts and four arms. Dense entries are bytes / test MAE (gallons per 100 miles); greedy and MILP entries are bytes / MAE ratio to that row's dense teacher. U→D: UNKNOWN with identical dense fallback; C→D: CPU cap with dense fallback. Thus all Local DP ratios equal 1 and its sizes equal Dense. No fallback is omitted or credited as compression.

n	Tol. (% IQR)	Anchors	Dense	Greedy	MILP	Local DP
24	1	Yes	566 / 0.5844	484 / 1.003	432 / 1.002	U→D
24	1	No	566 / 0.5844	474 / 1.001	476 / 0.995	U→D
24	2	Yes	566 / 0.5844	484 / 1.003	455 / 1.002	U→D
24	2	No	566 / 0.5844	474 / 1.001	487 / 1.002	U→D
24	5	Yes	566 / 0.5844	484 / 1.003	442 / 1.013	U→D
24	5	No	566 / 0.5844	442 / 1.012	350 / 0.995	U→D
48	1	Yes	581 / 0.5831	C→D	613 / 1.001	U→D
48	1	No	581 / 0.5831	C→D	586 / 0.998	U→D
48	2	Yes	581 / 0.5831	C→D	532 / 1.002	U→D
48	2	No	581 / 0.5831	C→D	398 / 1.009	U→D
48	5	Yes	581 / 0.5831	C→D	475 / 1.004	U→D
48	5	No	581 / 0.5831	C→D	349 / 0.988	U→D

Results and limits. All twelve Local DP searches are UNKNOWN with dense fallbacks and identical model hashes, hence no new sparse original-model witness. Greedy's six 24-label candidates use less storage within the error threshold, whereas all six 48-label searches reach their CPU limit. MILP returns a checked sparse witness on all twelve contracts; ten use fewer bytes within the threshold. Its two 48-label, 1%-tolerance candidates use 613 and 586 bytes, compared with 581 for the dense teacher, despite having fewer jumps. These baseline successes show that the consumer can reward sparse compatible functions, but they supply no Local DP advantage.

Prediction timing used five separate single-thread processes per candidate, each predicting all 398 feature values 1,000 times. Median batch CPU is approximately 0.020 seconds; identical dense and Local DP-fallback functions vary by up to about 5%. This prevents attributing small differences to algorithmic prediction speed. Rationalization error is at most 5.35×10^{-7} target units; interpolation of the rational teacher on the 24/48-label grid introduces maximum full-domain errors of 0.01543/0.00799, respectively. None of the 77 test inputs is outside the training domain. These approximation errors, feasibility checks, storage costs and predictive errors are distinct quantities.

A retained-trace construction diagnosis. A subsequent read-only audit inspects all 108 Local DP budget traces, without another solver call or test access. There are 30,339 endpoint records and 8,400 within-case keys; 7,195 keys recur across budgets. All 1,758 distinct complete endpoint certificates pass rational replay, including explicit binding to the positive/negative endpoint objectives. The original implementation starts a new endpoint cache for every budget. After a key first has a complete certificate, 8,824 later records charge 29,653 endpoint calls. These are potential reuse opportunities, not measured savings: the traces lack per-endpoint timings, and a changed resource-limited search may explore different states. Partial and UNKNOWN records cannot be reused as exact certificates. The audit preserves the original downstream negative and supplies no new predictive or storage result.

Construction-only cache comparison. A separately fixed comparison on the twelve retained contracts then tests reuse of complete exact certificates across budgets, without refitting or accessing test targets. All 24 method–contract searches reach $k = 0, \ldots, 8$ and use dense fallbacks. The cached implementation records 13,020 new solver calls versus 36,880 without reuse (median paired reduction 64.47%), but complete-process CPU rises from 988.59 to 1094.81 seconds and is higher in all twelve pairs. These implementation counts do not establish a sparse-model or timing benefit. The required rational replay passes 15 of 24 method–contract outputs (135 of 216 budget traces) before its fixed 600-CPU-second limit; nine outputs remain unverified and the prespecified positive-result criterion is not accepted. In the replayed subset, one conditional rejection becomes UNKNOWN with caching, rather than establishing an original-model rejection. This exposed-input diagnostic leaves the held-out downstream result unchanged; Table 22 gives every pair and its verification coverage.

Table 22: Complete construction-only cache comparison on the twelve retained contracts. U: uncached; C: cached. CPU includes the complete measured process. P: rational replay passed; ?: not reached within the fixed replay allowance. Every arm returns UNKNOWN with dense fallback. The call reductions coexist with higher CPU in every pair; partially verified outputs are not promoted to certified sparse models.

		Interior	New calls		Complete CPU (s)		Replay	
n	Width (%)	anchors	U	C	U	C	U	C
24	1	yes	3286	994	68.71	72.44	P	P
24	1	no	3260	1004	69.52	73.25	P	P
24	2	yes	2976	1050	73.93	77.86	P	P
24	2	no	2941	1079	80.87	87.07	P	P
24	5	yes	2898	1105	83.11	88.42	P	P
24	5	no	2963	1106	83.30	88.54	P	P
48	1	yes	3233	1113	85.92	93.66	P	P
48	1	no	3248	1114	85.87	94.03	P	?
48	2	yes	3106	1111	87.47	96.56	?	?
48	2	no	3177	1116	89.32	105.12	?	?
48	5	yes	2845	1112	89.58	100.47	?	?
48	5	no	2947	1116	90.99	117.38	?	?

O PROSPECTIVE STRUCTURED PANEL: VERIFICATION UNRESOLVED

Design, fixed before any method ran. The panel has 24 analytic inputs crossing $n \in \{12, 24, 48\}$, $\beta^+ \in \{2, 4\}$, two layouts of the two uncertain samples (at the boundary or in the interior) and two fresh seeds. Each seed is derived from a hash of the dispatch and cell identity, and no input was selected or filtered on outcomes. Each generator places eight events at slots $\lfloor rn/8 \rfloor$, $r = 0, \ldots, 7$, with rational amplitudes in $\{2, \ldots, 6\}/64$ and event times in $\{1/4, 3/8, 1/2\}$; $D = 1$, $J = 1/8$, all windows are $[0, 1]$, and uncertain boxes have half-width $1/1024$. Every input passed an exact feasibility check of its generator, and generator witnesses, seeds and structure metadata are withheld from all methods. Each input is queried at $k \in \{2, 4, 6\}$, and 288 executions (72 queries, four methods) run in a counterbalanced order fixed with the inputs.

Methods and limits. A–C share Bounds + block-cap, with the optional all-triples packing bound replaced by the typed valid bound $\mathrm{OPT} \geq 0$, a choice made from the two-input diagnosis after these inputs were fixed and before any method ran on them. A continues with exact enumeration, B with Local DP and then the enumeration fallback, and C with the activation MILP (accepting only exactly checked witnesses) and then the enumeration fallback; D is the standalone activation MILP. Each execution has 45 CPU seconds, a 90-second wall watchdog, 1,024 calls, one thread and 4 GiB, on a shared server pinned to one core. Eight outcome-blind rational controls and typed-zero mutation checks passed before scoring. Complete-process CPU runs from before process spawn and source import until after child reaping and evidence serialization, and is the parent plus reaped-child CPU of each execution.

Recorded outcomes and unresolved verification. Table 23 preserves every cell's producer-reported statuses. The retained report lists 72 Local-DP decisions (7 ACCEPT, 32 front-end rejections and 33 local exclusions), versus 36 for enumeration and 43 for the MILP cascade. Of A's 36 UNKNOWNs and C's 29, 27 each were stopped by CPU limits; other unresolved rows exhausted calls or certificates. D reports seven positive candidates and 65 numerical infeasibility statuses, the latter never treated as exact rejections. These are retained output counts, not newly verified evidence of comparative superiority.

The initial checker passed all 288 records but accepted stored Local-DP replay flags and incompletely bound some common-bound summaries. A strengthened, solver-disabled closeout inspects all 288 records and reports 137 row passes and 151 row failures: stored certified-upper/root-upper summaries disagree with its direct root-primal reconstruction. This prevents independent derivation of affected final decisions. All 40 Local-DP budget bindings match the supplied k, and all seven forged-evidence controls are rejected, but the global check fails. It uses 437.44 cumulative CPU seconds and 437.46 wall seconds, including the preserved initial path failure, within the original 600/900-second verification allowance. No new solver call or input is used.

The available evidence does not determine whether the mismatch lies in stored summaries or an overly strict reconstruction condition, and it establishes no wrong scientific answer. We therefore withhold the panel from the abstract and confirmatory comparisons; passing rows are not selectively promoted. The recorded call totals (1,513 Local DP, 40,896 enumeration and 33,758 MILP cascade) and timings remain implementation diagnostics, not verified scaling evidence.

Scope. The panel is not used to establish a certified advantage. It is prospective but small and synthetic, with only two uncertain samples per input; at $n \geq 24$ the producer reports only negative or unresolved outcomes. Its retained outputs emphasize exclusion attempts over feasible-query coverage at larger sizes. The server was shared, so timings are finite observations and not a clean-host benchmark.

P PROSPECTIVE LENGTH PANEL

Design, fixed before any solve. The panel uses the generator of Appendix H with parameters not used anywhere else in this work: $b \in \{4, 8, 16, 32\}$, $s \in \{1, 2\}$, $a \in \{21/32, 27/32\}$, $t = 3/8$ and $h = 13/32$. It is queried at $k = 2b - 1$, giving 16 inputs with $n = 15$ to 130 and $\beta = 2$. Every generator witness (with $2b$ events) was checked exactly before any solve and was withheld from the methods. The arms are the four Length methods, with the packing bound replaced by $\mathrm{OPT} \geq 0$,

Table 23: Retained producer statuses, not confirmed exact-decision counts: two seeds × three budgets = 6 queries per cell. FE: reported front-end rejections shared by A–C. B's reported ACCEPT count is in parentheses. D: reported positive candidates + numerical negatives. Capped: A / C executions stopped by the CPU limit.

n	β^+	Layout	FE	A: enum.	B: Local DP	C: MILP casc.	D: MILP	Capped
12	2	boundary	3	4	6 (1)	5	1 + 5N	0 / 0
12	2	interior	2	4	6 (2)	6	2 + 4N	0 / 0
12	4	boundary	4	4	6 (2)	6	2 + 4N	0 / 0
12	4	interior	2	3	6 (2)	5	2 + 4N	0 / 0
24	2	boundary	2	2	6 (0)	2	0 + 6N	4 / 4
24	2	interior	2	2	6 (0)	2	0 + 6N	4 / 4
24	4	boundary	4	4	6 (0)	4	0 + 6N	2 / 2
24	4	interior	2	2	6 (0)	2	0 + 6N	4 / 4
48	2	boundary	2	2	6 (0)	2	0 + 6N	4 / 4
48	2	interior	2	2	6 (0)	2	0 + 6N	4 / 4
48	4	boundary	3	3	6 (0)	3	0 + 6N	3 / 3
48	4	interior	4	4	6 (0)	4	0 + 6N	2 / 2

and the standalone activation MILP of the taxi and structured panels. The MILP's science code is byte-identical to that implementation; it is called through its core entry point because the wrapper's memory limit is unsupported on the host, and an unused spreadsheet dependency was stubbed. Limits are 45 CPU seconds, a 90-second wall watchdog, 512 calls and one thread per attempt. Four attempts ran concurrently on a heavily shared laptop (median CPU/wall ratio 0.83), so timings are indicative.

Verification. All 30 decisive rows were replayed without a solver. All 68 forged-evidence checks (dropped rows, changed budgets, forged witnesses) were rejected, and no two arms contradict each other. In addition, every supplied budget equals the pre-fixed value $2b - 1$ and matches the manifest written before the run. The panel used 982 child-CPU seconds in total.

Table 24: All 80 attempts of the prospective Length panel. Each cell gives the outcome and child CPU in seconds. A: certified ACCEPT; R: certified REJECT; U: UNKNOWN; N: numerical infeasibility only (not a REJECT).

ID	b	s	a	n	k	Bounds	+cap	+Local DP	+enum	MILP
PL-01	4	1	21/32	15	7	U 0.8	U 0.8	A 1.3	U 31.7	A 0.7
PL-02	4	1	27/32	15	7	U 0.8	U 0.8	A 1.2	U 31.8	A 0.7
PL-03	4	2	21/32	18	7	U 0.8	R 0.8	R 1.1	U 45.3	N 0.6
PL-04	4	2	27/32	18	7	U 0.9	R 0.9	R 1.1	U 45.4	N 0.7
PL-05	8	1	21/32	27	15	U 1.2	U 1.2	A 2.5	U 45.5	A 0.8
PL-06	8	1	27/32	27	15	U 1.2	U 1.2	A 2.5	U 45.5	A 0.9
PL-07	8	2	21/32	34	15	U 1.5	R 1.5	R 2.3	U 45.5	N 0.7
PL-08	8	2	27/32	34	15	U 1.5	R 1.4	R 2.3	U 45.5	N 0.7
PL-09	16	1	21/32	51	31	U 2.6	U 2.6	A 6.9	U 45.4	A 1.3
PL-10	16	1	27/32	51	31	U 2.5	U 2.5	A 6.8	U 45.4	A 1.4
PL-11	16	2	21/32	66	31	U 3.9	R 4.0	R 6.7	U 45.4	N 0.8
PL-12	16	2	27/32	66	31	U 3.8	R 3.9	R 6.7	U 45.4	N 0.8
PL-13	32	1	21/32	99	63	U 7.7	U 7.6	U 22.7	U 45.3	A 3.7
PL-14	32	1	27/32	99	63	U 7.7	U 7.7	U 21.1	U 45.3	A 3.5
PL-15	32	2	21/32	130	63	U 12.8	R 12.9	R 27.0	U 45.3	N 1.5
PL-16	32	2	27/32	130	63	U 12.5	R 12.5	R 26.4	U 45.4	N 1.5

Outcomes. Local DP returns 6 ACCEPT, 8 REJECT and 2 UNKNOWN. The two UNKNOWNs (PL-13/14, $n = 99$) stop after 510 of the 512 allowed calls. On the six accepted inputs the Local DP witness support equals the replayed integer relaxation lower bound (6, 12 and 24), so these are exact minima. The standalone MILP returns eight checked, non-minimal witnesses and eight numerical infeasibility statuses, all consistent with the certified rejections. Block-cap rejects the eight $s = 2$

inputs. Enumeration decides nothing: 13 of its attempts are stopped by the CPU limit and three end without a certificate.

Q LENGTH RERUN WITH THE PACKING BOUND REPLACED

Protocol. The rerun uses the 16 Length inputs, the four original methods, the supplied budgets $k = 2b-1$ and the original limits: 45 CPU seconds per child, a 90-second wall watchdog, 512 solver calls and one numerical thread. The only change is the one fixed by the packing ablation (Appendix L): the optional all-triples packing lower bound is replaced by the typed valid bound $\mathrm{OPT} \geq 0$. That bound can never reject a budget, and every REJECT rests on an original relaxation, block-cap, Local DP or enumeration certificate. The code is a byte-identical copy of the ablation code (all 22 source hashes match). The frozen input manifest was not available on the local host, so the 16 inputs were regenerated from the hash-identical study specification with the original generator. S1-09 is byte-identical to the ablation input, all 16 generator witnesses match the original study's generator checks, and the exported decisive witnesses of S1-01, S1-02, S1-05 and S1-06 are exactly feasible for the regenerated observations. A manifest was written before any solve.

Execution and verification. The rerun used a shared laptop with six concurrent single-threaded attempts and no CPU pinning, whereas the original study ran on a server; timings are therefore indicative, and short attempts include interpreter start-up. The 64 attempts used 592.6 child-CPU seconds in total. A separate verifier re-derived the inputs, checked every payload and trace hash, enforced that no trace entered the packing stage, and replayed every one of the 28 decisive answers with the original zero-solver replays while all solver entry points raised errors. All 66 forged-evidence mutations were rejected, and no input received contradictory decisions. Eight enumeration attempts with $b \in \{8, 16\}$ were stopped by the CPU limit inside enumeration and are UNKNOWN. This is a post hoc look at retained inputs, and it was not used to select any input or parameter of a prospective panel.

Table 25: All 64 rerun attempts (OFF) beside the original study (ON). Each cell is the outcome and child CPU in seconds; A: certified ACCEPT, R: certified REJECT, U: UNKNOWN. ON and OFF ran on different hosts.

						OFF (rerun)				ON (original)			
ID	b	s	a	n	k	Bounds	+cap	+DP	+enum	Bounds	+cap	+DP	+enum
S1-01	2	1	11/16	9	3	U 0.6	U 0.6	A 0.7	A 3.0	U 0.2	U 0.2	A 0.3	A 2.5
S1-02	2	1	13/16	9	3	U 0.5	U 0.6	A 0.7	A 2.9	U 0.2	U 0.2	A 0.3	A 2.5
S1-03	2	2	11/16	10	3	U 0.5	R 0.5	R 0.6	R 6.7	U 0.2	R 0.2	R 0.3	R 6.4
S1-04	2	2	13/16	10	3	U 0.5	R 0.5	R 0.6	R 6.7	U 0.2	R 0.2	R 0.2	R 6.3
S1-05	4	1	11/16	15	7	U 0.6	U 0.6	A 1.0	U 27.2	U 0.3	U 0.3	A 0.6	U 26.7
S1-06	4	1	13/16	15	7	U 0.6	U 0.7	A 0.9	U 27.2	U 0.3	U 0.3	A 0.6	U 27.0
S1-07	4	2	11/16	18	7	U 0.7	R 0.7	R 0.8	U 38.9	U 0.7	R 0.7	R 0.8	U 39.4
S1-08	4	2	13/16	18	7	U 0.6	R 0.6	R 0.8	U 38.9	U 0.7	R 0.7	R 0.8	U 39.8
S1-09	8	1	11/16	27	15	U 0.9	U 1.0	A 2.0	U 45.3	U 45.0	U 45.0	U 45.0	U 45.0
S1-10	8	1	13/16	27	15	U 0.9	U 0.9	A 2.1	U 45.3	U 45.0	U 45.0	U 45.0	U 45.0
S1-11	8	2	11/16	34	15	U 1.2	R 1.3	R 1.8	U 45.3	U 45.1	U 45.1	U 45.1	U 45.1
S1-12	8	2	13/16	34	15	U 1.2	R 1.3	R 1.8	U 45.3	U 45.1	U 45.1	U 45.1	U 45.1
S1-13	16	1	11/16	51	31	U 2.2	U 2.3	A 6.0	U 45.3	U 45.0	U 45.0	U 45.0	U 45.0
S1-14	16	1	13/16	51	31	U 2.3	U 2.3	A 6.1	U 45.3	U 45.0	U 45.0	U 45.0	U 45.0
S1-15	16	2	11/16	66	31	U 3.5	R 3.5	R 6.0	U 45.3	U 45.0	U 45.0	U 45.0	U 45.0
S1-16	16	2	13/16	66	31	U 3.5	R 3.5	R 5.9	U 45.3	U 45.0	U 45.0	U 45.0	U 45.0

Exact minima. S1-09 reproduces the ablation exactly: ACCEPT at $k = 15$ with 149 charged calls, a certified relaxation lower bound of 12 and a witness with 12 positive labels. On S1-09/10 and S1-13/14 the Local DP witness support equals the replayed integer relaxation lower bound (12 and 24), so these inputs have certified minima of 12 and 24, below the generating supports of 16 and 32. The Local DP REJECTs on S1-11/12/15/16 are exact exclusions through the supplied budget.

R CERTIFYING HEURISTIC EVENT COUNTS

Inputs and reference minima. The pilot uses all 32 Retail-I and 96 Retail-II inputs from their frozen input manifests. Each reference minimum is the least budget accepted by the original enumeration, with every smaller budget rejected; all 32 Retail-I minima match those listed in Appendix I.3. A manifest with the methods and all hyperparameters was written before any solve.

Proposers. (i) The ℓ_1 relaxation minimizes $\sum_i j_i$ subject to (1)–(3). (ii) Iteratively reweighted ℓ_1 runs five rounds with weights $1/(j_i + 10^{-3})$, fixed in advance. Both outputs are feasible; they are recovered as exact rational witnesses, and $\hat{k}$ counts the positive jumps of the exact witness. (iii) ℓ_1 trend filtering of the box centers minimizes $\sum_i |p_i - c_i| + \lambda \sum_i |p_{i+1} - 2p_i + p_{i-1}|$ over free positions for $\lambda \in \{0.01, 0.1, 1\}$, fixed in advance. Its count $\hat{k}$ is the number of kinks. The fit is rationalized at the LP vertex and checked exactly for box membership and convexity; when both hold, an exact fixed-position LP decides whether the model can realize the fit.

Classification and verification. A proposal is *exact* if $\hat{k} = \mathrm{OPT}$, an *over-count* if $\hat{k} > \mathrm{OPT}$ (the certificate exhibits a smaller certified budget), and *inconsistent* if $\hat{k} < \mathrm{OPT}$ or no compatible trajectory realizes the output. All 640 method–input records were recovered exactly; none is UNKNOWN or numerical-only, and exact and floating-point counts never disagree. A solver-free verifier rechecked every record in exact arithmetic and rejected all 12 forged variants.

Interpretation. Because $\sum_i j_i = w_n - w_0$, the ℓ_1 objective is the net velocity change and has many ties; on every input the optimal face also contains a solution with exactly OPT events, so the ℓ_1 over-count depends on which vertex HiGHS returns. Trend filtering with $\lambda < 1/4$ returns the box centers themselves, so its small-λ rows are identical. At $\lambda = 1$, 116 of the 118 inconsistent fits leave a box; the other two lie in the boxes but have a single kink of size $23/20$, which exceeds the cap $J = 1$, so their count of one is below $\mathrm{OPT} = 2$. Trend-filter kinks are not events: one jump inside an interval can create two kinks, and a large kink can require two jumps. The four release regimes of a window are correlated, so the counts are not prevalence estimates.