AUDITING STATIC OCCLUSION ORDERING UNDER PREDICTION-DEVIATION BUDGETS

Anonymous authors
Paper under double-blind review

ABSTRACT

After making a prediction, a data holder may want to release or store fewer fields while keeping that prediction approximately unchanged. One way to choose fields for deletion is to measure their individual effects once and try the least sensitive first. The difficulty is that every accepted deletion changes the context for later trials, and measuring sensitivities consumes queries that could instead explore another path. We audit this trade-off in a cost-tiered deletion procedure that checks each removal against the original prediction. On taxi and movie data, static sensitivity ordering reduces retained cost relative to alphabetical ordering and two random paths given the same query allowance. Under the baseline mask, its relative RMSE increase stays below the chosen tolerance over the first four tested budgets; wider budgets permit larger savings but can exceed that tolerance. Repeated random paths and cost reassignments largely preserve the advantage, whereas alternative masking can reverse it. Static sensitivities therefore provide useful ordering information in these settings, with the benefit depending on the allowed prediction drift and the representation of deleted fields.

1 INTRODUCTION

After obtaining a prediction, a data holder may want to release or store a reduced record while keeping that prediction approximately unchanged. Disclosure and feature-minimization work studies what to send or collect while preserving a model's usefulness (Zhou et al., 2026; Shaowang et al., 2025). We consider already collected fields: a trusted evaluator sees the complete input and fitted model, then minimizes the assigned cost of retained fields subject to a bound on prediction drift.

A natural rule tries fields with small individual prediction effects first. Those effects are measured on the complete row, however, and each accepted deletion changes the context for the next trial. The initial ranking may lose its value as deletion proceeds. Computing it also uses model queries. Are those queries better spent on a sensitivity ranking, or on trying another deletion path? Alphabetical ordering supplies a simple reference but cannot answer this second question.

We study a rule that scores each field once, visits cost tiers from highest to lowest, and tries less sensitive fields first within each tier. The scores set the order; a separate check accepts a deletion only when the current reduced row stays within the allowed deviation from the original prediction (Figure 1). We compare this rule with lexical ordering by field name and with a search that runs two random deletion paths and keeps the cheaper retained set.

The deviation budget controls how far deletion can move the prediction and therefore which paths are available. We sweep this budget in training-target standard deviations, so its scale is comparable across targets. At each budget we measure retained cost and error against observed targets. We also record how often the tested orders retain different sets, which shows where order has room to matter.

For the tested nonlinear predictors on taxi and movie data, the static ranking reduces retained burden relative to both comparators under the baseline mask. Its own prediction-error increase remains below the chosen reference over the first four budgets. Wider budgets produce larger savings, but the utility reference can then be exceeded. The comparison is also sensitive to how deleted fields are represented.

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.

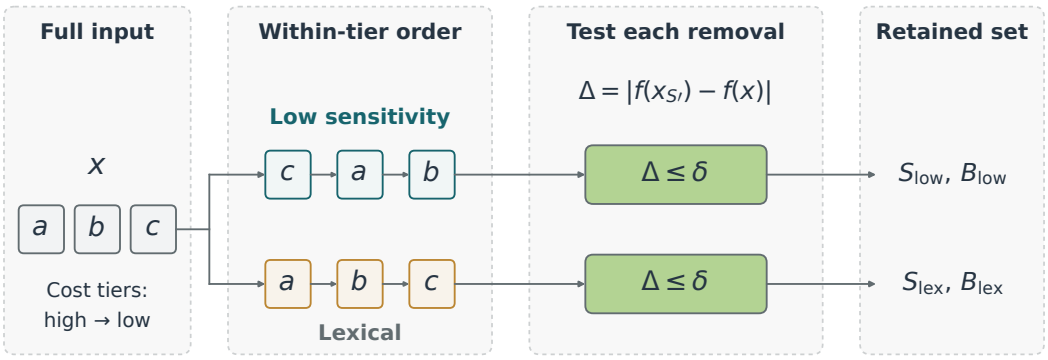

Figure 1: Two independent deletion paths with different within-tier orders. Both policies traverse cost tiers from high to low; the shown equal-cost fields follow the example in Section 4. Low-sensitivity scores are computed once per row on the full input; lexical ordering uses field names. For each field j, test $S' = S \setminus \{j\}$ against the fixed full-input prediction $f(x)$. If $\Delta \le \delta$, update $S \leftarrow S'$; otherwise retain the field without revisit, then continue to the next field. Outputs $S_{\text{low}}, S_{\text{lex}}$ are the final retained sets, and $B_{\text{low}}, B_{\text{lex}}$ their assigned retained burdens.

We make two contributions.

- **Ordering information during sequential deletion.** We show that fixed per-row sensitivity scores reduce retained burden beyond lexical ordering and the tested randomized search with the same model-query allowance.
- **Budget and masking dependence.** We relate the budget-dependent burden reduction to the method's own predictive utility and examine how masking choices change the ordering comparison.

2 Related Work

Data minimization and disclosure. Data minimization asks which information can be withheld while preserving a task's usefulness. Prior work searches privacy-ordered prompt transformations for language models (Zhou et al., 2026), trades user identifiability against prediction accuracy in sensor streams (Shaowang et al., 2025), and supplies tabular benchmarks for deidentification (Sen et al., 2023). For a fixed black-box predictor, Rastegarpanah et al. (2021) test whether a feature is individually necessary by replacing it with a constant across instances and measuring output changes. This motivates measuring individual effects, but our decision is sequential and specific to each row: which fields should remain after several deletions? We constrain prediction drift and measure assigned retained cost. A differential-privacy budget has a different meaning (Dwork et al., 2006; Kazan & Reiter, 2024).

Cost-sensitive feature acquisition. Feature acquisition begins with incomplete information and selects which values to obtain. Conditional-mutual-information methods learn a greedy selection rule (Covert et al., 2023) or estimate feature utility with non-uniform costs and per-sample budgets (Gadgil et al., 2024). Other approaches use a non-greedy oracle (Valancius et al., 2024), reinforcement learning (Kachuee et al., 2019), variational gates (Ardywibowo et al., 2022), or sensitivity under a denoising autoencoder (Kachuee et al., 2018). Closest to our use of rankings, Guney et al. (2025) train an acquisition policy from instance-specific local explanations. These methods motivate ordered, cost-sensitive selection. In our post-prediction setting the complete row is already available, so we can measure singleton effects directly and test a fixed ordering without training another policy.

Occlusion attribution and deletion order. Occlusion sensitivity measures the output change caused by masking an input component (Zeiler & Fergus, 2014). The removal operation is part of what an attribution method measures (Covert et al., 2021), which motivates our mask sensitivity

check. Deletion curves (Samek et al., 2017), deletion and insertion metrics (Petsiuk et al., 2018), and remove-and-retrain (Hooker et al., 2019) use ordered perturbations to evaluate explanations. Sufficient input subsets use instance-wise backward selection (Carter et al., 2019), while preferred abductive explanations find sufficient conditions for trees and forests (Audemard et al., 2022). Our study applies these ideas to retained cost under a fixed prediction-deviation budget. The empirical question is whether a once-only ordering remains useful during sequential deletion, including when its query allowance could instead fund random search.

3 Problem and Deployment Setting

The evaluator seeks a cheaper retained record while preserving the prediction already made from the complete row. The relevant constraint compares two outputs of the same fitted model; accuracy against the observed target is evaluated separately.

An example $x = (x_1, \ldots, x_d)$ contains d semantic fields with non-negative assigned weights r_j, and fields with equal weight form a tier. Encoded or derived columns belonging to one semantic field are removed atomically. A fixed predictor f maps a masked input to a scalar prediction, while x_S replaces every field outside retained set $S \subseteq \{1, \ldots, d\}$ with the domain's fixed mask representation.

The constrained reference problem is

$$S^*(x) \in \arg\min_{S \subseteq \{1,\ldots,d\}} \sum_{j \in S} r_j \quad \text{subject to} \quad |f(x_S) - f(x)| \leq \delta. \tag{1}$$

Equation (1) specifies the desired final subset. The procedure below searches a single deletion path, so its output is feasible but need not minimize this objective. Every trial uses the same full-input prediction as its reference; successive removals do not receive separate allowances that accumulate.

Budget and its scale. Because δ uses target units, a common absolute value represents different perturbation sizes across targets. We sweep a normalized budget $\tilde{\delta}$ and set each domain budget using the training target's standard deviation,

$$\delta_{\text{domain}} = \tilde{\delta}\, \sigma_{y,\text{domain}}, \tag{2}$$

Thus $\tilde{\delta}$ is dimensionless, while δ is an absolute difference in target units. This scaling lets us compare deletion under similar target-relative budgets without treating one year of IMDb production date as equivalent to one unit of log trip duration.

We evaluate an offline local simulation in which a data holder supplies a complete row to an evaluator holding the fitted model, and the evaluator returns a scalar prediction and a reduced record for release or storage. Full values, masked states, and intermediate outputs remain at the evaluator. When the data holder, model operator, and recipient are separate parties, this boundary requires a trusted evaluation site.

4 Static Occlusion-Ordered Traversal

Within a cost tier, every deletion saves the same assigned amount. We therefore try fields whose individual removal changes the full-input prediction least, then check whether each remains removable after earlier deletions. For field j, this initial effect is

$$s_j(x) = |f(x) - f(x_{-j})|, \tag{3}$$

where x_{-j} uses the same field-level mask as traversal. Compute these scores once per row, then sort each tier in ascending order, resolving ties by the fixed semantic-field order. The ranking remains unchanged as the retained set shrinks.

Ordering policies. All five orderings in Table 1 share semantic groups, descending-cost tiers, masks, and state-dependent feasibility checks, isolating within-tier order in the paired comparison.

Random uses seeded random keys within each tier; permuted-score ordering permutes a row's scores among fields before sorting. Model fitting and the auxiliary randomized paths introduce separate sources of variation.

Table 1: Original sweep: five within-tier orderings and their evaluation roles. The paired burden comparison uses the first two; all five enter the divergence and joint utility summaries.

Policy	Within-tier ordering	Study role	Paired R_B
Low sensitivity	Ascending $s_j(x)$	Focal	Yes
Lexical	Ascending field name	Comparator	Yes
Random	Seeded random keys	Auxiliary	No
High sensitivity	Descending $s_j(x)$	Auxiliary	No
Permuted score	Ascending permuted scores	Auxiliary	No

One-pass traversal. Each ordering follows the same one-pass deletion procedure:

1. Initialize $S \leftarrow \{1, \ldots, d\}$ over semantic groups and cache the full-input prediction $f(x)$.
2. Fix the within-tier order for the selected policy. For score-based orderings, compute singleton scores once from the full input and resolve sensitivity ties by ascending case-sensitive semantic-field name.
3. Traverse weight tiers from higher to lower assigned cost.
4. For each candidate j, construct $S' = S \setminus \{j\}$ with the same preprocessing and mask transformation, evaluate $f(x_{S'})$, and set $S \leftarrow S'$ when $|f(x_{S'}) - f(x)| \leq \delta$.
5. Retain each rejected field in S without revisiting it, and continue through the final candidate.
6. Return the retained set and its assigned burden. Evaluate the final reduced inputs to compute test RMSE.

This procedure assumes deterministic prediction for identical processed inputs. Each feasibility check uses the row after previously accepted removals, so whether a field can be deleted depends on what has already been deleted.

Figure 1 illustrates the effect of an early decision. Let $f(x) = 10.0$, $\delta = 1.0$, and let fields a, b, c each cost 2. Singleton predictions 10.8, 11.2 and 9.4 give low-sensitivity order c, a, b, versus lexical order a, b, c. Lexical traversal accepts a at 10.8, rejects b at 11.4, then accepts c at 10.2: it retains $\{b\}$ at cost 2. Low-sensitivity traversal accepts all three at 9.4, 10.2 and 10.9, leaving cost 0. Revisiting b would also complete the lexical path, but the rule permits no revisit. These predictions are illustrative.

Not revisiting fields limits traversal to at most d queries. Counting predictions per row, one full-input prediction, d singleton probes, at most d deletion trials, and the final RMSE prediction give an upper bound of $2d + 2$ evaluations for score-based ordering. A separate lexical or random traversal does not require singleton probes and uses at most $d + 2$ evaluations under the same accounting. Sorting costs $O(d \log d)$.

5 Evaluation Setup

At each budget, all deletion rules receive the same fitted model, test rows, costs and mask. We compare final retained burdens, then evaluate reduced-input predictions against observed targets. Sweeping only the budget tests how cost savings and target error change as more prediction drift is allowed.

Comparisons and query allowance. The original sweep compares the five orders in Table 1. Lexical order supplies the burden reference. In supplementary fits with the same data and model settings, Random $\times 2$ runs two independently randomized paths per row and returns the one with lower retained burden; ties use smaller prediction deviation, then the first path. It shares the original prediction between paths and uses at most $2d+2$ queries, matching static scoring under the counting convention above. This tests whether singleton probes are more useful than a second random path, without consulting target labels during selection.

Dynamic ordering is a further comparison: it refreshes candidate effects on the current reduced row, reuses candidate predictions, and still checks feasibility against the original prediction. It tests what is lost by keeping the scores fixed. The supplementary fits also support repeated random paths, cost reassignment and alternative masks. Their results are reported separately from the original sweep.

5.1 Evaluation metrics and reference criteria

Aggregate burden reduction. To measure savings, sum the costs of the fields each policy retains: $B_\pi(x) = \sum_{j \in S_\pi(x)} r_j$. Subscripts low and lex refer to low-sensitivity and lexical ordering. We compare their total burdens on the same complete test set $\mathcal{D}_{\text{test}}$:

$$R_B = 100 \, \frac{\sum_{x \in \mathcal{D}_{\text{test}}} B_{\text{lex}}(x) - \sum_{x \in \mathcal{D}_{\text{test}}} B_{\text{low}}(x)}{\sum_{x \in \mathcal{D}_{\text{test}}} B_{\text{lex}}(x)}. \tag{4}$$

All test rows contribute, including rows with zero lexical burden; total lexical burden is positive in the evaluated configurations. We compute the ratio per fit and budget, then average across fits. For other policies, replace B_{low} by B_π. Appendix F checks this estimator against the original summaries.

Utility degradation. A stable prediction can still be inaccurate, so we also compare reduced-input predictions with the observed targets. On n test pairs (x_i, y_i), let $\text{RMSE}_\pi = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f((x_i)_{S_\pi(x_i)}) - y_i)^2}$, and let $\text{RMSE}_{\text{full}}$ use the full inputs on the same pairs. For $\text{RMSE}_{\text{full}} > 0$, the relative change is

$$g_\pi = 100 \, \frac{\text{RMSE}_\pi - \text{RMSE}_{\text{full}}}{\text{RMSE}_{\text{full}}}. \tag{5}$$

Policy comparisons share nominal δ, while realized RMSE can differ and adjacent budgets reuse the same fitted models and test rows. The deviation constraint also gives a deterministic RMSE bound (Appendix D).

Small relative-error tolerances can summarize data-reduction trade-offs (de Oliveira et al., 2021). We use a 3% increase in RMSE as a descriptive reference. Focal-policy utility asks whether low-sensitivity ordering stays below it. The original sweep also reports a stricter joint reference: every one of the five policies in every one of the three fits must stay below 3%. A failure of this joint reference alone does not identify the focal policy as the cause.

Policy-divergence diagnostic. Order can affect the retained set without changing its cost. We therefore report set divergence alongside the burden contrast: the fraction of rows on which the five policies return at least two different retained sets. With Π denoting those policies and $n = |\mathcal{D}_{\text{test}}|$, this is

$$D_{\text{set}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[\exists \, \pi, \pi' \in \Pi : \; S_\pi(x_i) \neq S_{\pi'}(x_i)] \, . \tag{6}$$

Here $\mathbf{1}[\cdot]$ is the indicator function, and percentages report $100 D_{\text{set}}$. An auxiliary policy can create positive divergence even when the primary pair agrees. Different retained sets can also have equal burden.

Following the use of prevalence references in model-behavior audits (Pastor et al., 2021), we mark divergence of at least 5% as a descriptive reference. It indicates how often the tested orderings differ, without serving as a significance or power criterion.

5.2 Data, models, and budget settings

Domains and splits. The primary tasks predict NYC log trip duration and IMDb production year, each from 100,000 rows split 80,000/20,000. NYC uses January 2024 yellow-taxi trips in temporal order; IMDb retains the Join Order Benchmark source-file order. Carbon predicts grid intensity from 336 half-hour periods on 1–7 January 2024, with a chronological 268/68 split. Table 2 gives target scales; Appendix C supplies sources and preprocessing.

Table 2: Regression targets, fixed splits, and training-target standard deviations σ_y in target units. All orderings share the same rows within each domain.

Domain	Target	Row order	Train / test	σ_y
NYC	$\log(1 + \text{trip minutes})$	Temporal	80,000 / 20,000	0.6720
IMDb	production year	Source file	80,000 / 20,000	13.6435
Carbon	gCO_2/kWh	Chronological	268 / 68	53.9880

Models and implementation. We fit histogram-based gradient boosting (HGB), random forests (RF) and ridge regression (Ridge) in scikit-learn 1.5.2. Seeds 20260814–20260816 give three stochastic fits of HGB/RF; Ridge is deterministic. Models and fixed hyperparameters are shared across budgets and policies (Appendix C).

Fields, costs, and masking. NYC, IMDb and Carbon use $d = 7$, 6 and 4 numeric fields. Manual cost tiers are $(3, 1, 0)$ for NYC and $(3, 1)$ for IMDb/Carbon. Singleton probes and deletion trials replace fields by -1.0; models train on unmasked inputs. Table 7 lists every field and cost.

Budget grid. We use a geometric ladder symmetric in $\log_2$ about $\tilde{\delta} = 0.0744$, the NYC normalized operating point of the fixed-$\delta = 0.05$ evaluation:

$$\tilde{\delta} \in \{0.0186,\ 0.0372,\ 0.0744,\ 0.1488,\ 0.2976\} \text{ target-SD units.} \tag{7}$$

Only the budget changes, spanning a factor of sixteen. We set $\delta = \tilde{\delta}\,\sigma_y$ using full-precision training-target sample standard deviations (`ddof=1`); Table 2 rounds them for display.

5.3 Statistical aggregation

We compute one burden ratio per fit and budget on fixed test rows. Condition means and 95% bootstrap intervals use 5,000 resamples of the three fit-level values. A configuration is a domain, family, seed label and budget: 27 per budget, 135 overall. Ridge's labels share one deterministic fit, although random paths can vary. Supplementary contrasts use separate fits under the same settings, paired across policies.

6 Results

We first use the original budget sweep to relate lexical-relative savings to prediction error. We then ask whether the static scores outperform two random paths at the same query allowance, using the supplementary fits. This separation matters because the original utility summaries combine all five policies, while the supplementary comparisons also report each policy's own error.

6.1 NYC and IMDb budget sweep

Low-sensitivity ordering saves retained burden in every primary nonlinear NYC/IMDb condition over the first four budgets (Table 3). The mean advantage grows through $\tilde{\delta} = 0.1488$: from roughly 0.8–1.2% to 12–14% on NYC, and from roughly 0.3–0.4% to 3–4% on IMDb. Figure 2 shows the full ladder, including the smaller or sign-changing responses of Ridge and Carbon.

The five orders also retain different sets more often over these first four budgets. NYC divergence rises from 11–16% to 60–66%, and IMDb divergence from 4–6% to 37–45%. The lowest value is 3.7% for IMDb×HGB at $\tilde{\delta} = 0.0186$, where the burden contrast is nevertheless positive. Thus the 5% prevalence reference describes how frequently sets differ; it is not a requirement for a positive burden contrast.

At the widest NYC budget, divergence declines even as the mean burden advantage grows. The relationship between these measures is therefore budget dependent. Complete divergence values and native-unit budget conversions are in Table 6; we next examine the target-error cost of the additional deletions.

Table 3: Mean relative-burden reduction R_B (%) from Equation (4), comparing low-sensitivity with lexical ordering. Each entry gives the condition mean and a 95% bootstrap interval from three fits; † marks the condition with 3.7% any-policy set divergence.

Domain × family	$\tilde{\delta} = 0.0186$	0.0372	0.0744	0.1488
NYC × HGB	1.21 [0.94, 1.57]	2.92 [2.39, 3.67]	6.57 [6.41, 6.81]	14.18 [13.23, 15.69]
NYC × RF	0.82 [0.81, 0.85]	2.14 [2.05, 2.25]	5.44 [5.05, 5.92]	11.89 [11.19, 12.43]
IMDb × HGB	0.33 [0.32, 0.34]†	1.34 [1.09, 1.50]	2.83 [2.41, 3.06]	3.83 [2.41, 4.77]
IMDb × RF	0.39 [0.39, 0.40]	0.94 [0.87, 1.00]	2.10 [1.99, 2.17]	3.34 [3.31, 3.40]

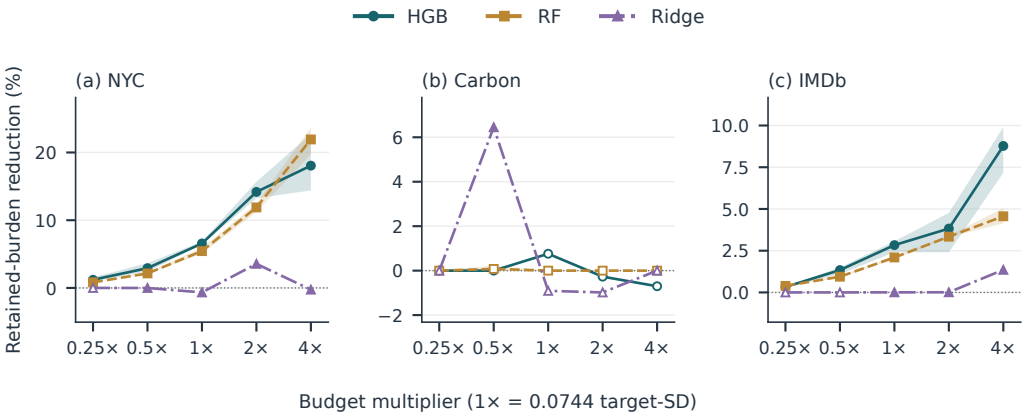

Figure 2: Assigned retained-burden reduction R_B (%) across normalized budgets. Bands show 95% bootstrap intervals from three fits; Ridge is deterministic. Filled markers indicate mean any-policy set divergence of at least 5%, and hollow markers indicate lower divergence. Utility counts are reported separately in Figure 3.

6.2 Burden and utility across budgets

In the original sweep, all policies and fits in the four primary nonlinear NYC/IMDb conditions satisfy the 3% relative-RMSE reference at the first four sampled budgets, ending at $\tilde{\delta} = 0.1488$, alongside the increasing burden reductions in Table 3. At the next sampled budget, $\tilde{\delta} = 0.2976$, mean reductions reach 18–22% on NYC and 5–9% on IMDb, but each primary condition fails this joint utility criterion. Figure 3A shows the burden contrasts together with the divergence and joint utility status; Appendix A reports the widest-budget values and intervals.

Table 4 and Figure 3C summarize individual configurations across all evaluated domains and models. Every policy and fit remains below the 3% relative-RMSE reference at the tested budgets up to $\tilde{\delta} = 0.0744$, with the largest measured change below 2.3%. At $\tilde{\delta} = 0.1488$, Carbon reaches 10.4%, while the four primary conditions continue to satisfy the joint utility reference. At $\tilde{\delta} = 0.2976$, the largest relative-RMSE change over all configurations and policies is 60.9%.

The counts separate two reasons for small practical differences. At the narrowest budget, only 9 of 27 configurations reach the divergence reference, while all meet the utility criterion. At larger budgets more orderings select different sets, but fewer configurations keep every policy below the error reference.

Ordering information at a matched query allowance. On the supplementary fits, static scoring gives higher R_B than Random ×2 in all 20 primary model–budget conditions and all 60 fit–budget comparisons. Table 5 shows the comparison at $\tilde{\delta} = 0.1488$, including each policy's largest RMSE change across fits. All twelve focal fits remain below 3% there, with a maximum of 2.257%; the same holds over the tighter budgets. At the widest budget, ten focal fits exceed the reference.

Table 4: Counts over 27 configurations per budget, including three seed labels for each deterministic Ridge fit. All-policy utility requires $g_\pi < 3\%$ for all five policies within a configuration. "All three criteria" additionally requires divergence of at least 5% and positive R_B; the final column is the largest relative-RMSE change over configurations and policies.

$\tilde{\delta}$	Configurations	Divergence $\geq 5\%$	All three criteria	All-policy utility $g_\pi < 3\%$	Worst relative RMSE change
0.0186	27	9	9	27	2.28%
0.0372	27	18	15	27	1.40%
0.0744	27	18	15	27	2.08%
0.1488	27	18	18	22	10.40%
0.2976	27	18	4	7	60.89%
Total	135	81	61	110	

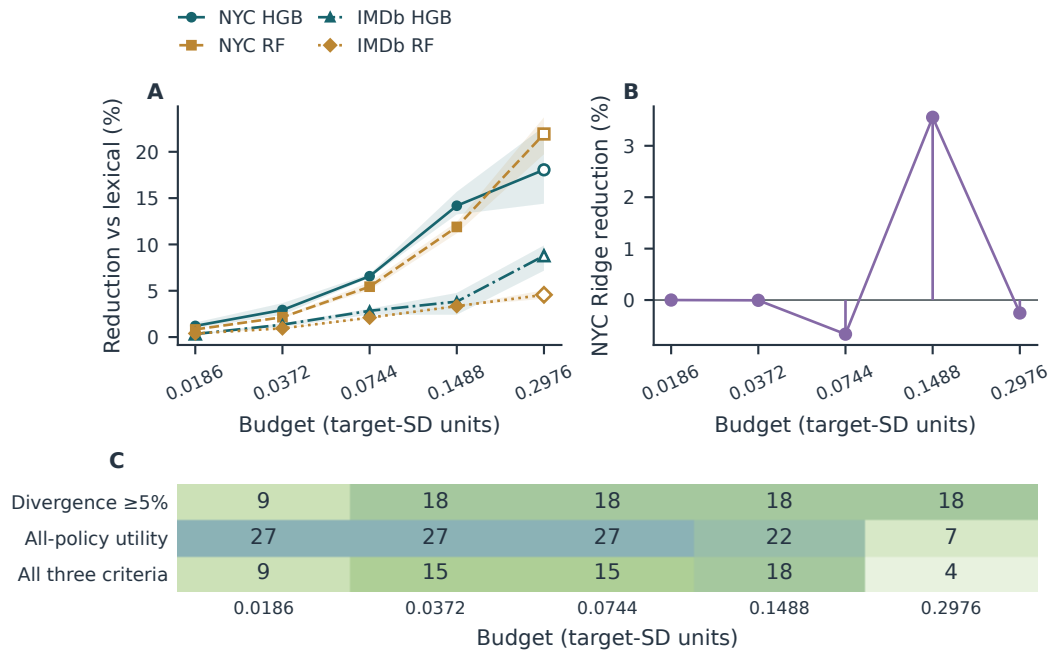

Figure 3: Burden and utility across the sampled budgets. (A) Nonlinear R_B with 95% bootstrap bands; hollow markers indicate mean divergence below 5% or failure of the joint utility criterion across all policies and fits. (B) Deterministic NYC Ridge. (C) Configuration counts from Table 4: all-policy utility requires $g_\pi < 3\%$ for all five policies; "All three criteria" also requires divergence of at least 5% and positive R_B.

Table 5: Supplementary ordering comparisons at $\tilde{\delta} = 0.1488$. Burden entries are mean R_B (fit-level SD); $g_{\max}$ is the largest signed relative-RMSE change across three fits. All values are percentages. Every column uses the same supplementary fits described above.

	R_B: mean (SD)				$g_{\max}$	
Domain / model	Static	Random	Random ×2	Dynamic	Static	Lexical
NYC HGB	14.18 (1.32)	-13.77 (5.86)	3.87 (3.10)	14.94 (1.08)	0.15	1.05
NYC RF	11.89 (0.64)	-6.96 (0.35)	6.16 (0.12)	15.87 (0.59)	2.26	2.55
IMDb HGB	3.83 (1.25)	-3.56 (0.77)	1.39 (0.19)	5.17 (0.92)	0.81	0.82
IMDb RF	3.34 (0.05)	-0.43 (0.07)	2.20 (0.07)	4.17 (0.04)	2.05	1.99

Refreshing the scores improves mean burden reduction in 18 of 20 conditions. Dynamic ordering uses 15 and 14 queries per NYC and IMDb row, compared with static scoring's 16 and 14. Static scoring therefore has no measured query saving over dynamic updating in these small tiers; its matched-query advantage is over the tested random search.

Sensitivity to paths, costs and masks. Repeating Random $\times 2$ with 50 path pairs per fit leaves every condition mean in favor of static ordering. Reassigning costs ten times while preserving tier sizes gives 199/200 advantages against each of lexical and Random $\times 2$. Replacing the sentinel mask with training medians and categorical modes preserves 19/20 and 18/20 condition-mean advantages, respectively. The latter comparison reverses for NYC HGB at $\tilde{\delta} = 0.1488$ and 0.2976. These checks preserve the fits and test rows, isolating choices within this evaluation; Appendix F gives the utility changes and complete settings.

7 Discussion

The matched-query comparison shows that static sensitivities can remain useful as deletion changes the row: singleton probes save more retained burden than two random paths. Dynamic ordering often saves still more and has no larger reported query count in these small tiers. The evidence therefore favors static scores over the tested random allocation, while exposing the benefit of refreshing them.

The budget determines which reduced inputs the procedure can accept, and the mask determines the predictions on those inputs. These choices jointly shape the apparent value of an ordering. A single budget or mask would miss both the growth in burden savings over the tighter grid and the observed reversals under alternative masking.

The deviation constraint bounds possible RMSE change (Appendix D); actual degradation also depends on the full predictor's errors. Measuring target RMSE alongside burden therefore remains necessary even when every deletion is feasible.

8 Limitations

The evidence uses fixed scalar-regression splits, with a small Carbon diagnostic. Ridge and Carbon show substantial model and domain dependence (Appendices E and B). Three-fit intervals describe variation on these rows, not over new datasets or splits.

Costs are manual weights, and masks are inference-time replacements applied to models trained on complete inputs. A replacement can place a row outside training support, and the alternative mask changes some policy rankings. The one-pass procedure guarantees its stated prediction constraint, with no approximation guarantee for the minimum-cost subset. Deployment additionally requires a trusted evaluator with access to the complete row and model.

9 Conclusion

For the tested NYC and IMDb nonlinear predictors under the baseline mask, static sensitivities reduce retained burden beyond lexical traversal and two random paths with the same query allowance. The first four budgets combine these savings with a focal RMSE increase below the chosen reference. Wider budgets and alternative masks change the trade-off. Initial field effects are therefore useful ordering information here, whose value depends on the permitted prediction drift and deletion representation.

STATEMENT ON THE USE OF LARGE LANGUAGE MODELS

OpenAI large language models, including Codex, were used for language polishing and to improve the clarity and readability of the manuscript. The authors take full responsibility for the scientific content, accuracy, and integrity of this work.

REFERENCES

Randy Ardywibowo, Shahin Boluki, Zhangyang Wang, Bobak J. Mortazavi, Shuai Huang, and Xiaoning Qian. VFDS: Variational foresight dynamic selection in bayesian neural networks for efficient human activity recognition. In *AISTATS*, 2022. URL https://proceedings.mlr.press/v151/ardywibowo22a/ardywibowo22a.pdf.

Gilles Audemard, Steve Bellart, Louènas Bounia, Frédéric Koriche, Jean-Marie Lagniez, and Pierre Marquis. On preferred abductive explanations for decision trees and random forests. In *IJCAI*, 2022. doi: https://doi.org/10.24963/ijcai.2022/91. URL https://www.ijcai.org/proceedings/2022/0091.pdf.

Brandon Carter, Jonas Mueller, Siddhartha Jain, and David Gifford. What made you do this? understanding black-box decisions with sufficient input subsets. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 567–576, 2019. URL https://proceedings.mlr.press/v89/carter19a.html.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021. URL https://jmlr.org/papers/v22/20-1316.html.

Ian Connick Covert, Wei Qiu, Mingyu Lu, Na Yoon Kim, Nathan J White, and Su-In Lee. Learning to maximize mutual information for dynamic feature selection. In *ICML*, 2023. URL https://proceedings.mlr.press/v202/covert23a/covert23a.pdf.

Jose Genario de Oliveira, Jr., Vipul Dhingra, and Christoph Hametner. Feature extraction, ageing modelling and information analysis of a large-scale battery ageing experiment. *Energies*, 14(17): 5295, 2021. doi: 10.3390/en14175295. URL https://www.mdpi.com/1996-1073/14/17/5295.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, 2006. doi: 10.1007/11681878_14. URL https://doi.org/10.1007/11681878_14.

Soham Gadgil, Ian Connick Covert, and Su-In Lee. Estimating conditional mutual information for dynamic feature selection. In *ICLR*, 2024. URL https://openreview.net/pdf/d27c0b9e152ca6e70d6be8b8d3402b85e83dad3e.pdf.

Osman Berke Guney, Ketan Suhaas Saichandran, Karim Elzokm, Ziming Zhang, and Vijaya B Kolachalama. Active feature acquisition via explainability-driven ranking. In *ICML*, 2025. URL https://raw.githubusercontent.com/mlresearch/v267/main/assets/guney25a/guney25a.pdf.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/fe4b8556000d0f0cae99daa5c5c5a410-Abstract.html.

Mohammad Kachuee, Sajad Darabi, Babak Moatamed, and Majid Sarrafzadeh. Dynamic feature acquisition using denoising autoencoders. In *TNNLS*, 2018. doi: 10.1109/tnnls.2018.2880403. URL https://arxiv.org/pdf/1811.01249.

Mohammad Kachuee, Orpaz Goldstein, Kimmo Kärkkäinen, Sajad Darabi, and Majid Sarrafzadeh. Opportunistic learning: Budgeted cost-sensitive learning from data streams. In *ICLR*, 2019. URL https://openreview.net/pdf/eb9189687175bb57690f0a8fa707f245189c0d77.pdf.

Zeki Kazan and Jerome P. Reiter. Prior-itizing privacy: A bayesian approach to setting the privacy budget in differential privacy. In *NeurIPS*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a47f5cdff1469751597d78e803fc590f-Paper-Conference.pdf.

Eliana Pastor, Luca de Alfaro, and Elena Baralis. Looking for trouble: Analyzing classifier behavior via pattern divergence. In *Proceedings of the 2021 International Conference on Management of Data*, pp. 1400–1412, 2021. doi: 10.1145/3448016.3457284. URL https://doi.org/10.1145/3448016.3457284.

Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized input sampling for explanation of black-box models. In *British Machine Vision Conference*, 2018. URL https://arxiv.org/abs/1806.07421.

Bashir Rastegarpanah, Krishna Gummadi, and Mark Crovella. Auditing black-box prediction models for data minimization compliance. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ac6b3cce8c74b2e23688c3e45532e2a7-Abstract.html.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 2017. doi: 10.1109/TNNLS.2016.2599820. URL https://iphome.hhi.de/samek/pdf/SamTNNLS17.pdf.

Aniruddha Sen, Christine Task, Dhruv Kapur, Gary Howarth, and Karan Bhagat. Diverse community data for benchmarking data privacy algorithms. In *NeurIPS*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a15032f8199511ced4d7a8e2bbb487a5-Paper-Datasets_and_Benchmarks.pdf.

Ted Shaowang, Shinan Liu, Jonatas Marques, Nick Feamster, and Sanjay Krishnan. Algorithmic data minimization for machine learning over internet-of-things data streams. In *VLDB*, 2025. URL https://www.vldb.org/pvldb/vol18/p5652-shaowang.pdf.

Michael Valancius, Maxwell Lennon, and Junier Oliva. Acquisition conditioned oracle for nongreedy active feature acquisition. In *ICML*, 2024. URL https://raw.githubusercontent.com/mlresearch/v235/main/assets/valancius24a/valancius24a.pdf.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pp. 818–833, 2014. doi: 10.1007/978-3-319-10590-1_53. URL https://cs.nyu.edu/~fergus/papers/zeilerECCV2014.pdf.

Jijie Zhou, Niloofar Mireshghallah, and Tianshi Li. Operationalizing data minimization for privacy-preserving llm prompting. In *ICLR*, 2026. URL https://openreview.net/pdf/c45241837ef3452393e092a489c464e00f0234b7.pdf.

A WIDEST-BUDGET RESULTS

At $\tilde{\delta} = 0.2976$, the nonlinear reductions are $+18.06\%$ [14.40, 22.71] for NYC×HGB, $+21.92\%$ [19.73, 23.73] for NYC×RF, $+8.77\%$ [7.16, 9.90] for IMDb×HGB, and $+4.56\%$ [4.13, 5.05] for IMDb×RF; brackets give 95% bootstrap intervals from three fits, and all four conditions fail the all-policy, all-fit utility criterion. On IMDb×Ridge, the measured reductions are $+0.005\%$ at 0.0744, $+0.01\%$ at 0.1488, and $+1.35\%$ at 0.2976. The reference is also exceeded at 0.1488 on Carbon RF and Ridge, and at 0.2976 on NYC HGB/RF, IMDb HGB/RF, and Carbon HGB/RF/Ridge; 25 of 135 configurations fail the all-policy criterion.

Table 6: Native-unit budget conversions and any-policy set divergence across the five policies. The upper block maps normalized budgets to target units; the lower block gives the mean percentage of rows with differing retained sets across three fits.

$\tilde{\delta} = 0.0186$	0.0372	0.0744	0.1488	0.2976
Budget in target units				
NYC (log minutes)	0.0125	0.025	0.05	0.10	0.20
IMDb (years)	0.25	0.51	1.02	2.03	4.06
Carbon (gCO_2/kWh)	1.0	2.0	4.0	8.0	16.1
Any-policy set divergence (%)				
NYC × HGB	15.6	29.3	53.7	59.9	33.7
NYC × RF	11.1	22.1	43.0	65.9	49.8
NYC × Ridge	0.3	14.1	45.3	64.3	61.0
IMDb × HGB	3.7	12.6	29.4	44.9	41.9
IMDb × RF	6.4	13.3	25.2	36.6	49.4
IMDb × Ridge	1.4	2.2	13.0	17.0	15.0
Carbon × HGB	4.4	0.0	4.4	1.5	2.9
Carbon × RF	0.5	0.5	0.0	0.0	0.0
Carbon × Ridge	0.0	50.0	4.4	2.9	0.0

B THE CARBON DOMAIN

The Carbon domain comprises 336 half-hour periods of grid carbon intensity, with a chronological split containing 268 training and 68 test rows. Both HGB and RF remain below the 5% any-policy set-divergence reference at every budget, with mean prevalence between 0 and 4.4%, corresponding to at most 3 of 68 rows on average across the three fits. The RF predictor yields identical burden and RMSE across the five policies at four budgets and $+0.08\%$ burden reduction at 0.0372.

For HGB, R_B reaches $+0.8\%$ at 0.0744, then changes to -0.3% and -0.7% at the two wider budgets, each with divergence below 5%. Ridge crosses the divergence reference only at 0.0372 ($+6.45\%$, with mean divergence of 50%, or 34 of 68 rows on average), while both neighboring budgets remain below the divergence reference. Carbon RF and Ridge exceed the relative-RMSE reference from 0.1488 upward.

In the fixed-$\delta = 0.05$ evaluation, Carbon's normalized budget is about one eightieth of NYC's. Carbon also remains largely order-invariant on the normalized grid, showing that matching target-relative budgets does not ensure that orderings produce distinct retained sets.

C SETTINGS OF THE ORIGINAL BUDGET SWEEP

Data and software. NYC uses the TLC file `yellow_tripdata_2024-01.parquet` from `https://d37ci6vzurychx.cloudfront.net/trip-data/`. IMDb uses `title.csv` from `https://event.cwi.nl/da/job/imdb.tgz`. Carbon uses daily responses from `https://api.carbonintensity.org.uk/intensity/date/` for 1–7 January 2024. The execution environment is Python 3.12.13, NumPy 2.1.3, pandas 2.2.3, PyArrow 18.1.0, and scikit-learn 1.5.2. The thread limits `OMP_NUM_THREADS`, `OPENBLAS_NUM_THREADS`, `MKL_NUM_THREADS`, and `NUMEXPR_NUM_THREADS` are all 1; `PYTHONHASHSEED=0`.

Row selection and preprocessing. For NYC, we read `tpep_pickup_datetime`, `tpep_dropoff_datetime`, and the six raw columns for the other inputs in Table 7, dropping rows missing any of them. Duration is the dropoff-minus-pickup timestamp difference in minutes. We retain $1 <$ duration < 120 and $0 <$ trip distance < 100, stably sort by pickup timestamp, and select the first 100,000 rows. The target is $\log(1 + \text{duration})$.

For IMDb, the expected column order is `id`, `title`, `imdb_index`, `kind_id`, `production_year`, `imdb_id`, `phonetic_code`, `episode_of_id`, `season_nr`, `episode_nr`, `series_years`, and `md5sum`. Rows with fewer than 12 columns, non-integer production year or kind ID, or a year outside $[1900, 2025]$ are skipped. Blank or unparsable series IDs, season numbers, and episode numbers become 0.0; blank titles become empty strings. Processing stops after the first 100,000 valid rows in source order.

For Carbon, periods lacking actual or forecast intensity are skipped, and the `from` timestamp supplies the calendar fields. The target is actual intensity. The chronological split is 268/68, as in Table 2. The original execution audit records 336 parsed periods; its staged JSON differs byte-for-byte from the historical frozen serialization. The reported parsed-row SHA-256 is `87818c8f85c3408c96873ec1aefcd5da12d2e88ea175671d80e7b53ffd759318`.

Fields and assigned costs. Table 7 lists every model input, its construction, and its assigned cost. These names also define lexical tie-breaking. The real-field tier sizes are $(3, 3, 1)$ for NYC at costs $(3, 1, 0)$, $(3, 3)$ for IMDb at $(3, 1)$, and $(2, 2)$ for Carbon at $(3, 1)$. NYC's zero-cost trip distance still participates in deletion trials and can affect feasibility even though it contributes no retained burden.

Table 7: Complete model-input specification. Rows within each domain follow model-column order. Costs are manually assigned; d excludes the evaluation-only ignored control.

Domain	Field name	Construction	r_j
NYC	`pickup_time`	Pickup weekday $\times 1440$ + hour $\times 60$ + minute	3
	`pickup_zone`	Numeric `PULocationID`	3
	`dropoff_zone`	Numeric `DOLocationID`	3
	`passenger_count`	Raw numeric passenger count	1
	`vendor`	Numeric `VendorID`	1
	`ratecode`	Numeric `RatecodeID`	1
	`trip_distance`	Raw numeric trip distance	0
IMDb	`kind`	Numeric `kind_id`	1
	`series_link`	$\log(1 +$ `episode_of_id`$)$	3
	`season`	Numeric `season_nr`	3
	`episode`	Numeric `episode_nr`	3
	`title_length`	Python character count of title	1
	`title_digit_count`	Count of title characters satisfying `str.isdigit()`	1
Carbon	`day_of_week`	Timestamp weekday	1
	`hour`	Timestamp hour	3
	`minute`	Timestamp minute	1
	`operator_forecast`	API forecast intensity	3

The evaluator appends `ignored_positive_control` with cost 3 to each field list, but not to the model matrix. It is automatically removed without a model query or budget check, so its retained-set bit is always zero and it adds no final burden. Thus the evaluator lists contain 8, 7, and 5 entries for NYC, IMDb, and Carbon, while d counts 7, 6, and 4 real inputs. Recorded deletion counts include this control.

Mask transformation. Each model input occupies a floating-point column in the representation described above. Categorical codes retain their numeric values, and continuous inputs retain their constructed scales. Missing values are handled through the stated row filters and IMDb zero replacements. Singleton probes and sequential deletions replace the selected value by -1.0 within the existing columns, without adding missingness indicators. The predictor is fitted to the unmasked training matrix and held fixed during deletion. Each accepted deletion is checked against the original full-input prediction, and the final reduced-input prediction supplies test RMSE.

Interpretation of costs and masks. The costs in Table 7 are manual experimental assignments shared across policies. The original sweep varies the deviation budget with these assignments and the sentinel fixed. Appendix F examines cost reassignments and an alternative mask; the assigned costs remain experimental weights rather than empirically calibrated measures. Masking is an inference-time intervention and can place reduced inputs outside the support of the unmasked training data. Deletion acceptance uses prediction deviation, with no additional training-support filter, and test RMSE evaluates predictive utility on the resulting inputs.

Model parameters and randomness. HGB uses `max_iter=120`, `learning_rate=0.06`, `max_leaf_nodes=15`, `l2_regularization=1.0`, and `random_state=seed`. RF uses `n_estimators=120`, `min_samples_leaf=5`, `max_features=0.8`, and `random_state=seed`; the budget sweep sets `n_jobs=1`. Ridge uses `alpha=2.0`. There is no hyperparameter search. The three seed labels are 20260814, 20260815, and 20260816. Each policy evaluation initializes NumPy's `default_rng` with `seed+1009*policy_index`, using the policy's index in the evaluator. Random orders fields by random keys within a tier; permuted-score ordering permutes the row's sensitivity vector before applying low-sensitivity sorting.

Budget precision. The normalized grid is computed as $\tilde{\delta} = 0.05m/0.6719956595807753$ for $m \in \{0.25, 0.5, 1, 2, 4\}$. The training-target standard deviations, computed with `ddof=1`, are 0.6719956595807753 for NYC, 13.64347789988714 for IMDb, and 53.988018802186595 for Carbon. At $m = 1$, the corresponding native budgets are 0.05, 1.0151462814803467, and 4.016991630263433. Fitted models and test rows are shared across the five budgets within each condition.

Aggregation. For each stochastic domain–family–budget condition, the interval uses 5,000 bootstrap resamples of its three fit-level mean reductions. Ridge contributes one distinct fitted model under three seed labels; the auxiliary randomized paths can vary across labels. Utility counts in Figure 3C and Table 4 refer to individual configurations, whereas a condition satisfies the joint utility reference only when all its configurations do. A condition with only some passing configurations contributes to the former count without satisfying the latter criterion.

Scope of the evaluation. The 3% relative-RMSE and 5% divergence references are descriptive criteria examined on the evaluation rows. IMDb retains source-file order, tying the reported performance to that split rather than to a temporal holdout. Equation (4) compares total retained burdens on all test rows per fit and budget, then averages across fits. The original joint criterion covers all five policies and fits; separate policy-specific utility for the reconstruction is reported in Appendix F.

D RMSE Bound Under a Deviation Constraint

The reduced and full predictions differ by at most δ on every evaluated row. Their prediction vectors therefore differ in Euclidean norm by at most $\sqrt{n}\delta$. Applying the reverse triangle inequality to their error vectors gives, on the same rows and with $\mathrm{RMSE}_{\mathrm{full}} > 0$ for the relative bound,

$$|\mathrm{RMSE}_\pi - \mathrm{RMSE}_{\mathrm{full}}| \le \delta, \qquad |g_\pi| \le 100\,\frac{\delta}{\mathrm{RMSE}_{\mathrm{full}}}, \tag{8}$$

This bound controls the largest possible absolute RMSE change. Whether deletion improves or worsens RMSE depends on its direction relative to the original prediction errors.

E Ridge Budget Sweep

Ridge exhibits non-monotone sign changes on NYC and small positive reductions on IMDb, as summarized in Table 8. Because Ridge is deterministic for these data and settings, each condition contributes one distinct fitted-model value.

Table 8: Deterministic Ridge R_B values (%) across the five sampled budgets. Positive values favor low-sensitivity ordering over lexical ordering.

Domain × family	$\tilde{\delta} = 0.0186$	0.0372	0.0744	0.1488	0.2976
NYC × Ridge	+0.00	−0.01	−0.66	+3.55	−0.25
IMDb × Ridge	0.00	0.00	+0.005	+0.01	+1.35

F Supplementary Ordering Comparisons

Evaluation. The supplementary comparisons use separately fitted models under the documented NYC and IMDb HGB/RF settings, all three seeds, five budgets and 20,000 test rows per fit. Input-file hashes match the original audit. Python is 3.12.0; the four library versions in Appendix C are unchanged. All comparisons here use the new fits. Equation (4)'s ratio of total burdens matches the original primary means within 0.0033 percentage points; averaging per-row ratios does not. The original numerical summaries are retained.

Comparators. Random ×2 selects the lower-burden output of two independent random traversals per row, breaking ties by prediction deviation and then the first path, without using target labels. Generators use the fit seed plus 1009×2 and 1009×5 over real fields. Sharing the full-input prediction gives the same $2d + 2$ query allowance as static scoring under Section 4's counting convention. Dynamic ordering recomputes effects on the current reduced input, reuses candidate predictions, checks feasibility against the full-input prediction, and never revisits rejected fields.

Results and utility. Static ordering has larger R_B than random and Random ×2 in all 20 primary model–budget conditions and all 60 fit–budget comparisons. At $\tilde{\delta} = 0.1488$, all twelve focal fits remain below 3%, with a maximum of 2.257%; at the widest budget, ten exceed the reference. Dynamic ordering improves the mean in 18 of 20 conditions, using 15 and 14 queries per NYC and IMDb row versus static scoring's 16 and 14. These small tiers therefore show no static query saving over dynamic updating. Budgets reuse models and test rows.

One joint flag differs from the original: at $\tilde{\delta} = 0.1488$, reconstructed NYC HGB high-sensitivity ordering reaches 3.048%, while the focal rule reaches at most 0.149%. The original joint summaries and these per-policy measurements describe separate runs. The supplementary CSV supplies all budgets and fit-level burdens and RMSE, including auxiliary orders and exhaustive final-subset search with its distinct query budget.

Sensitivity checks. Using the same fitted models and test rows, we repeated Random ×2 with 50 independently seeded path pairs per fit and reassigned each domain's costs ten times while preserving the cost multiset and tier sizes. All condition means favor static ordering over the random repeats; under cost reassignment, 199/200 comparisons favor it against each of lexical and Random ×2. Replacing the sentinel with training medians for numeric fields and modes for categorical codes preserves 19/20 and 18/20 condition-mean advantages against lexical and Random ×2, respectively. The latter reversals occur for NYC HGB at $\tilde{\delta} = 0.1488$ and 0.2976; at the widest budget its mean R_B is −2.82%. At $\tilde{\delta} = 0.1488$, the alternative mask leaves 11/12 static fits below 3%, compared with 12/12 under the sentinel. Tightening the tolerance to 2% under the sentinel reduces the count to 7/12. Random repetitions and cost assignments reuse fits and rows; the supplementary files provide complete outcomes and cost/mask settings.