SHARED THRESHOLDS FOR CHECKING DESCENT IN DISCRETE ROBUST LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

A training update that lowers a continuous relaxation can increase the discrete robust risk it approximates. We study how to verify a decrease in this risk while avoiding a full discrete solve at every update. For conditional value-at-risk (CVaR), the average loss in the worst-performing tail, a threshold optimized for one data subset also bounds the risk of other subsets. Sharing thresholds can therefore establish descent without solving each subset separately. We characterize how many thresholds suffice, bound the expected queries needed to find them under exact responses, and give smooth examples requiring queries proportional to the number of candidate subsets. Our algorithm, BRANCHCHECK, first checks one stored subset and a bound over all subsets, then searches further if needed. It accounts for numerical error and uses an exact solver when the bounds remain inconclusive. Experiments cover linear predictors, fixed nonlinear features, and a network trained in all layers, with all checking costs included. On Bike Sharing with 12×12 groups, greedy checking is 1.78 and 1.94 times faster than direct dynamic programming for the linear and neural models, with identical learning paths. The first subset check and global bound provide most of the saving. Further search improves the linear result but has no clear additional benefit for the network: its value depends on whether avoided exact solves repay the search cost.

1 INTRODUCTION

Robust learning can evaluate a model by its worst tail risk over permitted subsets of data groups. These subsets may obey a deletion budget, group quotas, or row-and-column constraints. A continuous relaxation supplies update directions, but a lower relaxed objective need not imply a lower discrete risk: two smooth CVaR examples have the same relaxed decrease and opposite exact changes (Theorem 1). The relaxation uses one tail threshold for all subsets, while each subset has its own optimal threshold. We study training rules that require every accepted update to lower this finite-sample risk. The question is computational: when can verifying the required decrease cost less than solving the full discrete problem?

Optimization layers and combinatorial or nonsmooth differentiation provide learning signals (Amos & Kolter, 2017; Vlastelica et al., 2020; Bolte et al., 2021; Grazzi et al., 2024). Structural conditions ensure binary feasibility or preserve minima under single-variable binary flips (Huang et al., 2026; Banerjee et al., 2026), while rounding and annealing improve discrete-solution recovery (Wang et al., 2022; Ichikawa, 2024). Distributional and outlier robust optimization (DORO) and interval-CVaR address outliers (Zhai et al., 2021; Liu & Pang, 2022), group distributionally robust optimization (group-DRO) optimizes subpopulation and tail risks (Soma et al., 2025), and federated group DRO uses a scalar threshold to reduce communication (Guo & Yang, 2024). Our task is to check a proposed update. Branch-and-bound can stop once upper and lower bounds determine whether the objective decreases (Land & Doig, 1960). We study how sharing thresholds across retained collections can reach that decision sooner.

CVaR's variational form (Rockafellar & Uryasev, 2000; Zhu & Fukushima, 2009) gives the key observation: a threshold minimizing one collection's risk also upper-bounds every other collection's risk at the same parameters. For each collection, thresholds giving a bound below the current risk form an interval. A set intersecting every interval establishes descent, even when no single threshold

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.

works for all collections (Figure 1). A classical interval result (Gyárfás & Lehel, 1985) determines the smallest such set. Finding and applying these thresholds is a separate computational task.

BRANCHCHECK begins by evaluating a previously stored feasible collection. It shares the resulting threshold and bounds groups of collections together through a search tree. If those bounds leave the update unresolved, it searches further within a fixed budget and then uses an exact solver when necessary. Every decision accounts for numerical error. The practical question is whether the saved exact solves outweigh the cost of these additional bounds.

Our contributions are as follows.

- We show that identical relaxed information can conceal opposite changes in a smooth discrete risk, and characterize threshold sets sufficient for descent.
- We bound discovery queries under exact responses, fixed minimizing thresholds, and strict descent below the current lower bound. We distinguish discovery from verifying supplied thresholds, and give worst-case query, tree-computation, and numerical-accuracy bounds.
- Complete training comparisons give DP/greedy time ratios of 1.78 and 1.94 on linear and neural Bike 12×12 models, with identical learning paths. Further search improves on the first check for the linear model, while the network's paired saving interval spans zero. Direct DP wins on smaller public-data families.

2 CHECKING FINITE-STEP DESCENT

2.1 STRUCTURED DELETION AND THE PROPOSED UPDATE

Partition the data into G groups of common size s_g. Let $\mathcal{B}$ be the permitted retained collections. Each $B \in \mathcal{B}$, called a branch, contains k groups and $m = ks_g$ observations; $i \in B$ indexes these observations. The family can impose a deletion budget, disjoint quotas, or one group per row and column of a grid. To measure performance on the largest losses within a branch, let θ be the model parameters, $\ell_i(\theta)$ the loss on observation i, and $R(\theta)$ a regularization penalty. The tail mass $\beta \in (0, 1)$ specifies the fraction of losses to average. A threshold $t \in \mathbb{R}$ separates losses contributing to this tail, with $[z]_+ = \max\{z, 0\}$. The CVaR objective at a chosen threshold is

$$F_B(\theta, t) = t + \frac{1}{\beta m} \sum_{i \in B} [\ell_i(\theta) - t]_+ + R(\theta). \tag{1}$$

Minimizing over t averages the largest βm losses, with a fractional boundary observation when necessary. Each branch chooses its own minimizing threshold to obtain risk f_B. Taking the worst of these risks gives the discrete objective V_{D}. Requiring all branches to use one threshold instead gives the relaxation V_{H}:

$$f_B(\theta) = \min_t F_B(\theta, t), \qquad V_{\mathrm{D}}(\theta) = \max_{B \in \mathcal{B}} f_B(\theta), \qquad V_{\mathrm{H}}(\theta) = \min_t \max_{B \in \mathcal{B}} F_B(\theta, t). \tag{2}$$

Thus $V_{\mathrm{H}} \geq V_{\mathrm{D}}$: one common threshold cannot improve on separate thresholds. The interval and query analysis applies to any continuous $F_B(\theta, t)$ that is convex in t and grows without bound as $|t| \to \infty$, with no convexity requirement on θ. This permits nonlinear predictors. CVaR-specific bounds and solvers use the loss structure above. At fixed θ, grouped CVaR admits a mixed-integer linear program (MILP). Relaxing discrete retention vectors to their convex hull gives V_{H}, which proposes updates.

To check a proposed update, we compare bounds on the risk before and after it. The baseline bounds on the current risk satisfy $V_{\mathrm{D}}(\theta) \in [L_0, U_0]$. A direction d and step size $\eta > 0$ propose displacement $s = \eta d$ and new parameters $y = \theta + s$, called the endpoint. The required change is

$$\Delta_{\mathrm{D}}(\theta, s) = V_{\mathrm{D}}(y) - V_{\mathrm{D}}(\theta) < 0. \tag{3}$$

One branch can establish nondecrease, whereas descent requires an upper bound covering all branches. Let $\ell_B \leq f_B(y)$ be a branch lower bound, $L_1 = \max_B \ell_B$, and $U_1 \geq V_{\mathrm{D}}(y)$ a global upper bound. Comparing these with the current bounds gives

$$U_1 < L_0 \implies \Delta_{\mathrm{D}} < 0, \qquad L_1 \geq U_0 \implies \Delta_{\mathrm{D}} \geq 0. \tag{4}$$

The second test identifies a branch whose risk reaches or exceeds the current upper bound. If neither test holds, the result is *unknown*. Update directions may come from exact or approximate relaxed solves, including valid first-order directions at nonsmooth points.

In training, a margin $\gamma > 0$ sets the minimum required improvement: acceptance requires $U_1 < L_0 - \gamma$. To see how bound accuracy affects this test, let $a = V_{\mathrm{D}}(\theta) - V_{\mathrm{D}}(y)$ be the true decrease, e_0 the uncertainty below the current risk, and e_1 the uncertainty above the proposed risk. Then

$$a - \gamma > e_0 + e_1, \qquad e_0 = V_{\mathrm{D}}(\theta) - L_0, \quad e_1 = U_1 - V_{\mathrm{D}}(y). \tag{5}$$

The improvement beyond γ must therefore exceed the combined uncertainty. This includes incomplete optimization and numerical error. A decrease of at most γ cannot pass even with exact values; wider bounds can also leave larger decreases unresolved. The sign results below use $\gamma = 0$; a positive margin shifts the cover level to $L_0 - \gamma$.

2.2 What relaxed information leaves unresolved

To identify what a relaxed decrease misses, define the relaxation gap $\Gamma = V_{\mathrm{H}} - V_{\mathrm{D}}$. The true change equals the relaxed change minus the change in this gap:

$$\Delta_{\mathrm{D}} = V_{\mathrm{H}}(y) - V_{\mathrm{H}}(\theta) - \{\Gamma(y) - \Gamma(\theta)\}. \tag{6}$$

If the relaxed decrease exceeds the initial gap, then $V_{\mathrm{H}}(y) < V_{\mathrm{D}}(\theta)$ establishes descent immediately. For a smaller relaxed decrease, the sign also depends on the endpoint gap. The following construction gives opposite smooth risk changes under the same relaxed information. Its minimax weights combine branches attaining the relaxed maximum. Here x is a scalar model parameter, $\Delta_{\mathrm{H}}(\theta, s) = V_{\mathrm{H}}(\theta + s) - V_{\mathrm{H}}(\theta)$ is the relaxed change, and superscripts $+$ and $-$ label the two instances.

Theorem 1 (Identical relaxed information, opposite updates)**.** *For every $\delta > 0$, there are two group-deletion CVaR problems with two groups of two observations and tail mass $3/4$ whose relaxed values both equal $10\delta - x$ on $[0, \eta]$, where $\eta = \delta/4$. At 0 they have the same relaxed minimizer, minimax weights, variational response values and first derivatives at that minimizer, and gap δ. The negative relaxed gradient $d = -V'_{\mathrm{H}}(0) = 1$ gives*

$$\Delta^{\pm}_{\mathrm{H}}(0, \eta) = -\eta < 0, \qquad \Delta^{+}_{\mathrm{D}}(0, \eta) = \eta, \qquad \Delta^{-}_{\mathrm{D}}(0, \eta) = -\eta.$$

Both discrete values are differentiable throughout the update segment.

The difference comes from an observation below that threshold: its derivative disappears from the relaxed response but enters the exact branch risk while the active branch remains fixed. The smooth construction in Theorem 4 also preserves every initial observation derivative.

3 Sharing thresholds across branches

3.1 Shared thresholds turn branch checks into a cover

We can reuse a threshold because $f_B(y) \le F_B(y, t)$ for every branch B and every t. Let T be a finite nonempty set of evaluated thresholds, called the threshold pool. Each branch uses the tightest bound available in T, denoted $u_B(T)$; their maximum $U_1(T)$ bounds the full risk. The useful thresholds for branch B form the set $I_B(L_0)$:

$$u_B(T) = \min_{t \in T} F_B(y, t), \qquad U_1(T) = \max_{B} u_B(T), \qquad I_B(L_0) = \{t : F_B(y, t) < L_0\}. \tag{7}$$

The sublevel interval I_B contains the thresholds whose branch upper bound is below the current lower bound. One threshold can lie in several intervals, resolving several branches at once. When the pool intersects every interval, $U_1(T) < L_0$ and the update decreases the discrete objective.

Theorem 2 (Geometry of shared threshold bounds)**.** *Assume the finite family $F_B(y, \cdot)$ is continuous, convex, and coercive. Then:*

1. *$V_{\mathrm{D}}(y) < L_0$ if and only if every $I_B(L_0)$ is nonempty.*
2. *$V_{\mathrm{H}}(y) < L_0$ if and only if $\bigcap_B I_B(L_0)$ is nonempty.*
3. *When all intervals are nonempty, the smallest cardinality of a threshold set with $U_1(T) < L_0$ equals the largest number of pairwise disjoint intervals among $\{I_B(L_0)\}$.*

The first two equivalences follow from the branch minima and minimax order. The third is the interval-piercing identity, where piercing means selecting points that intersect every interval (Gyárfás & Lehel, 1985): successively choosing a point just before the smallest right endpoint matches the disjoint-interval lower bound. Write $P(y, L_0)$ for this minimum number of thresholds, called the cover size. The algorithm must also find such thresholds without knowing the intervals in advance.

A running endpoint. In Figure 1, branch 213 retains columns 2, 1, and 3 in rows 1, 2, and 3. With current risk 10.625, $\gamma = 0$, and $T = \{0\}$, separate bounds leave 123, 213, and 321 unresolved. Evaluating 123 supplies threshold 9 and leaves only 213 unresolved. The root, which represents all branches, still has bound 11, so search splits it. Evaluating 213 then supplies threshold 5, reducing the combined tree bound to $10.5 < 10.625$. With no search budget, the unresolved root instead invokes dynamic programming (DP). Section C.5 gives the full calculation.

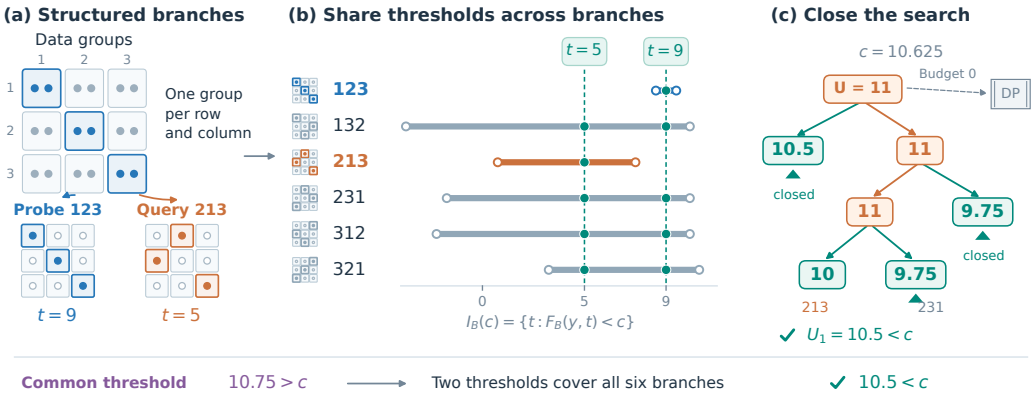


Figure 1: Shared-threshold descent checking. (a) Each branch retains one group per row and column; probes of 123 and 213 supply thresholds 9 and 5. (b) These thresholds intersect every strict sublevel interval; hollow endpoints are excluded. (c) After probing 123, tree upper bounds guide pruning; triangles mark closed subtrees and budget zero invokes DP. The best common threshold gives $10.75 > c = 10.625$, while the shared pool gives $U_1 = 10.5 < c$. Section C.5 and Figure 4 give the exact losses and search trace.

3.2 Uniform discovery and its query bound

A *branch query* evaluates one branch risk and returns a minimizing threshold. With numerical error, it returns a containing interval and a bound on threshold suboptimality. A *threshold sweep* evaluates $F_B(y, t)$ for every branch at the same t, also with error bounds when needed. Their counts are Q_b and Q_s. Each query either identifies a branch proving nondecrease or supplies a threshold that tightens the remaining upper bounds.

For initial valid upper bounds u_B^0, the branches needing information are

$$\mathcal{C}(y, L_0) = \{B : u_B^0 \geq L_0\}, \qquad K = |\mathcal{C}(y, L_0)|. \tag{8}$$

At most K exact queries and K sweeps give a separated decision (Theorem 5). Nondecrease needs one branch reaching the current risk, potentially found last. Strict descent requires bounding every

branch below the comparison level. The next result exploits sharing for strict descent, with exact responses and fixed minimizing thresholds. Here $\mathbb{E}$ averages over random selection and $\delta \in (0,1)$ is the allowed failure probability.

Theorem 3 (Discovering a small threshold cover)**.** *Assume $V_{\mathrm{D}}(y) < L_0$, $K \geq 1$, and exact sweeps. Each branch query returns a minimizing threshold t_B chosen by a fixed rule. Let $P_{\mathcal{C}}$ be the smallest number of thresholds intersecting all intervals $I_B(L_0)$ for $B \in \mathcal{C}$. With an unlimited budget, uniform selection in* BRANCHCHECK *satisfies*

$$\mathbb{E}Q_b \leq \min\left\{K,\ 2P_{\mathcal{C}}\left[1+\log\frac{K}{P_{\mathcal{C}}}\right]\right\}, \qquad Q_s \leq Q_b. \tag{9}$$

The sweeps count excludes initial screening. The algorithm needs neither $P_{\mathcal{C}}$ nor the interval endpoints. Its high-probability cost is $O(P_{\mathcal{C}}\log(eK/P_{\mathcal{C}}) + \log(1/\delta))$.

Choosing a unique minimizing threshold. The rule $t_B \in \arg\min_t F_B(y,t)$ is deterministic and independent of query history. Any such rule is valid. Writing $\ell_{(j)}$ for the jth largest loss, our CVaR routine returns $\ell_{(\lceil \beta m \rceil)}$. At integral $\beta m = j < m$, it selects the upper endpoint of the minimizing interval $[\ell_{(j+1)}, \ell_{(j)}]$.

Why the cover controls discovery. A useful threshold can remove many unresolved branches at once. In the proof, intervals sharing one point are grouped by the side on which their minimizing thresholds lie. Convexity makes a query remove every earlier threshold in that ordering. Uniform selection therefore removes a substantial part of such a group in expectation, yielding the logarithmic query bound. Section A.5 gives the full argument, including ties. In the running example the full cover has $P = 2$, while screening after the probe leaves $K = P_{\mathcal{C}} = 1$. The logarithm is attained on nested weighted-CVaR intervals, with P groups of n intervals, $K = Pn$, and $\mathbb{E}Q_b = PH_n$, where $H_n = \sum_{j=1}^{n} 1/j$; a specified greedy order can use K queries at $P = 1$. Response overestimate ϵ and threshold suboptimality ζ replace the cover level by $L_0 - \epsilon$, provided descent exceeds $\epsilon + \zeta$. The tail bound, matching example, and numerical extension appear in Section A.5, Proposition 16, and Corollary 7.

4 FROM QUERY BOUNDS TO COMPUTATION

4.1 WHAT INFORMATION A SHARED QUERY REVEALS

To ask whether sharing can always avoid examining many branches, we count the endpoint information an algorithm must request. An oracle model specifies which information is free and which calls are charged. Public input $\mathcal{I}_0$ comprises the family, the step, all initial observation and branch values and derivatives, curvature bounds, the exact current value and gap, and the relaxed solution and gradient trajectory. Endpoint calls are

$$\mathsf{Branch}(B) = (f_B(y), t_B), \qquad \mathsf{Sweep}(t) = (F_B(y,t))_{B\in\mathcal{B}}.$$

Write $Q_{\mathrm{branch}} = Q_b$ and $Q_{\mathrm{sweep}} = Q_s$. Each call costs one unit. Given independent internal randomness, each adaptive call and the final decision depend on $\mathcal{I}_0$ and the preceding call types, arguments, and responses. Sweeps may use any real threshold. This model measures acquired endpoint information. Runtime on explicit data also includes model evaluation and the operations used to combine branch bounds.

Here M counts the alternatives to distinguish, L bounds gradient change per unit displacement, and η is the step length.

Theorem 4 (Linear mixed-query lower bound)**.** *For every $M \geq 1$, $L > 0$, and $\eta > 0$, there is a group-deletion CVaR family with $M+2$ groups of three observations, tail mass $5/6$, and one retained group. Observation losses are convex and smooth; risk branches are L-smooth on $[0,\eta]$. All instances share initial observation and branch values and derivatives, exact baseline, scalar gap, and relaxed value and gradient trajectories. The step follows the initial negative-gradient direction and decreases V_{H} by $L\eta^2/48$. The initial set in Equation (14) has $K = M$. One instance decreases V_{D}; each of M alternatives increases it. Any algorithm deciding the sign with error at most $\epsilon < 1/2$ on every instance satisfies*

$$\mathbb{E}[Q_{\mathrm{branch}} + Q_{\mathrm{sweep}}] \geq (1-2\epsilon)M \tag{10}$$

on the decreasing instance. An exact deterministic decision requires at least M such queries.

Disjoint threshold intervals. Each of the M alternatives changes responses within its own threshold interval. A branch query or sweep can examine at most one such interval. Distinguishing an alternative from the decreasing instance requires examining its interval with probability at least $1 - 2\epsilon$. Summing gives the bound; Section A.6 constructs the smooth losses and proves the claim. □

Numerical separation. Errors of at least $L\eta^2/12$ can hide the sign on this family. The stated guarantee requires $(1 - 2\epsilon)M$ more accurate calls. If accuracy $\tau > 0$ costs τ^{-p}, where $p > 0$ controls cost growth, matching work bounds have order $M(L\eta^2)^{-p}$ (Corollary 8). Numerically, wider bounds can therefore increase the unresolved family.

4.2 Checking many branches through a search tree

From one branch to a search tree. The first branch evaluation, called the probe, tests whether one shared threshold suffices. It reuses branch B_* from the latest accepted lower bound or exact solve. Index selection costs $O(1)$ and quantile evaluation costs $O(m)$ after computing losses; all are timed. Methods share this rule and a four-branch threshold cache (Section C.4).

If the first threshold leaves too many branches unresolved, a tree lets us bound whole subfamilies without listing them. For matching retention, a tree node fixes a row prefix and the allowed columns of its next row. Singleton choices are absorbed; other column sets split into two fixed balanced subsets. With r remaining rows and current allowed set S, its descendant count is $|S|(r-1)!$. Let $\mathcal{B}_N$ be the branches below node N. At each threshold we maximize over this subfamily, then keep the tightest bound across T:

$$U_N(T) = \min_{t \in T} \max_{B \in \mathcal{B}_N} F_B(y, t).$$

For an $n \times n$ grid, the inner maximum is a maximum-weight assignment: each cell is weighted by its sum of losses above t. It costs $O(n^3)$ using cached cell scores. Cardinality and disjoint-quota families use selection instead. Each node records which thresholds it has processed, so generating N_{gen} nodes requires at most $N_{\text{gen}}|T|$ assignments. This bound can be looser than $\max_{B \in \mathcal{B}_N} \min_{t \in T} F_B(y, t)$ because every branch within a node uses the same threshold.

Numerical decisions must allow for both evaluation error and an approximately minimizing threshold. If $\widehat{F}$ is the computed response, e bounds its absolute numerical error, and ζ bounds threshold suboptimality, then $[\widehat{F} - e - \zeta, \widehat{F} + e]$ contains the branch risk. Shared scores round upward and use exact integer assignments. Acceptance requires $U_1 < L_0 - \gamma$. Equality or an unsafe numerical range leaves the comparison unresolved; a failed exact solve is reported as a failed run (Section E.1).

Greedy search processes the node with the largest stored upper bound. Uniform search uses random priorities that select unresolved branches uniformly, without enumerating them; Section C.4.1 gives the full queue algorithm. A zero expansion budget still evaluates the probe and root.

Algorithm 1 Checking a proposed update

Require: Proposed parameters y, current bounds $[L_0, U_0]$, required decrease γ, initial thresholds T, stored branch B_*, expansion budget b

1: Set acceptance level $c = L_0 - \gamma$; compute endpoint losses and numerical error bounds; return unknown if a numerical range check fails
2: Evaluate B_*; return nondecrease if its lower bound is at least U_0
3: Add its minimizing threshold to T and bound the full family; return descent if this bound is below c
4: Start a search tree containing the full family
5: **while** unprocessed nodes remain and fewer than b internal nodes have been expanded **do**
6: Select a node by the greedy or uniform rule and update its bound using T
7: Close it if its upper bound is below c; otherwise split it if it contains multiple branches
8: For an unresolved single branch, evaluate its risk and share its minimizing threshold; return nondecrease if its lower bound is at least U_0
9: **end while**
10: Combine bounds from evaluated branches, closed nodes, and remaining nodes; return descent if $U_1 < c$, otherwise unknown

For a binary tree of depth at most h, let K_0 count branches left unresolved by one retained initial threshold. Search expands at most hK_0 internal nodes. With exact fixed minimizing thresholds and unlimited budget, uniform tree search has the same branch-query distribution as flat discovery, so the bound in Equation (9) applies. Theorem 9 and Section A.8 prove these claims and their numerical extension. Unresolved checks invoke the exact solver and refine current-risk bounds if needed. Training accepts a verified decrease or halves the step and retries. Accepted bounds become the next baseline; cached thresholds are reevaluated at each endpoint. Timing includes setup, model evaluation, all checks, rejected trials, fallback, and baseline refinement (Section C.4). Checking two supplied thresholds can be computationally hard even when one newly queried threshold establishes descent (Theorem 15). Finding a useful threshold and verifying an existing set have different costs.

5 Experiments

We test whether relaxed descent can increase the discrete risk, and when threshold sharing saves computation. All studies use regression with half squared prediction error and regularizer $R(\theta) = 0.015\|\theta\|^2$. Predictors are linear models, linear readouts of fixed random Fourier features (RFF), and a trainable network with rectified linear units (ReLU). Exact comparison values come from enumeration, dynamic programming (DP), or MILP with numerical error bounds. Each CPU worker uses one numerical/solver thread; checking methods run in shuffled order. Reported 95% Student intervals use independent data seeds, or subsampling/initialization seeds conditional on the public datasets.

5.1 Relaxed decrease and shared information

The matched-proposal study gives every checker the same updates on twelve groups of four observations and six model parameters. It varies deletion count, tail mass, and whether retained groups must be balanced across two halves (Section C). Twenty seeds, three path positions, and six proposal scales give 6,480 endpoints. Enumeration finds 3,361 decreases and 3,119 nondecreases. Among 3,432 relaxed decreases, 124 increase the exact objective. Greedy and uniform sharing correctly decide every endpoint (Figure 2).

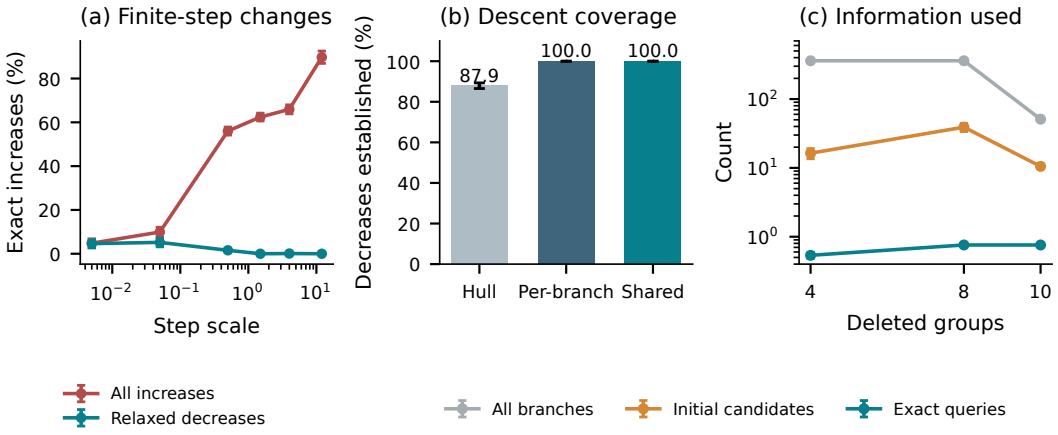


Figure 2: Matched proposal decisions. Left: exact increases across step scales. Middle: descent coverage using an endpoint relaxed solve (Hull), separate branch evaluations, or shared thresholds. Right: branch queries and initial candidates. Means and 95% intervals use 20 data seeds.

Evaluating the current relaxed threshold at the endpoint leaves 21.98 candidates from 257 branches on average. Greedy, its no-sharing ablation, and uniform use 0.685, 0.891, and 0.768 branch queries. The endpoint relaxed solve establishes 2,954 of the 3,361 decreases. These results isolate the information gained by sharing thresholds; runtime comparisons also charge the work needed to apply them.

5.2 PUBLIC DATA AND THE SIMPLE-PROBE BASELINE

Bike Sharing uses 2011 for training and 2012 for testing, with month by two-hour windows forming a 12×12 grid, or coarser 6×6 calendar groups. Appliances Energy splits at 15 April 2016 and uses Monday–Saturday by four-hour windows (Fanaee-T, 2013; Candanedo, 2017). Each feasible retained collection contains one group from every row and column; thus a 12×12 grid has 144 available groups and retains twelve. Models predict standardized $\log(1 + y)$ for observed response y; preprocessing uses training data only. Full heldout metrics include all test rows, including Energy Sundays. A fixed chronological subset measures worst-case tail risk (Section D.1).

The four checking policies share predictors and proposed updates. Direct DP solves the complete discrete problem at each trial. Probe/root evaluates the stored branch and the tree root, then falls back to DP if needed. Greedy and uniform additionally permit 64 internal-node expansions, differing in which unresolved branch they seek next. All use a decrease margin of 10^{-8}, at most 100 training iterations, and up to twelve backtracking trials per iteration; after stopping, recorded positions repeat the final model without further work (Section C.4). Across 360 linear trajectories, including two relaxed-acceptance controls, all descent-checking rules follow the DP parameter path. Bike 12×12 takes 2.372 seconds with DP, 1.507 with probe/root, and 1.335 with greedy. The paired savings are 1.037 ± 0.157 seconds against DP and 0.172 ± 0.104 against probe/root, with a DP/greedy mean-time ratio of 1.78. The probe contributes about 83% of the total saving. Probe/root is faster than uniform, and DP leads on the two small public-data families (Table 2).

A further 720 trajectories hold Bike's observation set fixed and vary linear features and fixed RFF maps of 128 or 512 cosine features (Rahimi & Recht, 2007). At 12×12, greedy's DP/greedy mean-time ratios are 1.78, 1.66, and 1.30. These models train the readout alone; Section D.2 reports their feature-preparation costs and matched-path comparisons.

5.3 A NETWORK WITH TRAINABLE NONLINEAR FEATURES

The width-16 ReLU multilayer perceptron (MLP) trains both weight matrices and biases: 433 parameters on Bike and 577 on Energy. Bike retains the same 1,728 observations at 6×6 and 12×12; Energy uses 432. Ten seeds, two tail masses, and four policies give 240 public-data trajectories, with another 320 across synthetic grids 6–12. The initial step is 0.02; the iteration limit, backtracking, acceptance margin, and search settings are shared with the linear comparison (Section D.3).

Data/grid	N	Direct DP	Probe/root	Greedy	Uniform
Bike 6×6	1,728	0.485 ± 0.021	0.510 ± 0.052	0.544 ± 0.063	0.546 ± 0.063
Bike 12×12	1,728	1.560 ± 0.083	0.757 ± 0.270	0.805 ± 0.321	0.819 ± 0.352
Energy 6×6	432	0.150 ± 0.036	0.205 ± 0.065	0.262 ± 0.075	0.281 ± 0.112

Table 1: Complete MLP training time in seconds, including model setup, forward and backward passes, all search work, fallback, and baseline refresh. N counts training observations. Entries average tail masses within seed, then ten seeds with 95% interval half-widths. All four policies have identical parameter and proposal paths.

All 560 trajectories preserve the same paths across policies, and all four parameter blocks change during training. On Bike 12×12, greedy saves 0.755 ± 0.239 seconds against DP, with a 1.94 ratio of mean times. Probe/root takes 0.757 seconds and greedy 0.805; their paired saving, greedy relative to probe/root, is -0.048 ± 0.062 seconds. Together with the linear results, this locates the benefit of sharing and the model-dependent return from additional search.

5.4 WHERE SEARCH PAYS FOR ITS COST

On the Bike network, further search saves 0.025 seconds in fallback DP but raises shared-bound and search costs from 0.023 to 0.108 seconds (Figure 3). Table 1 excludes common data loading and solver-environment setup. Adding the same 0.01287 seconds to each unrounded mean gives DP, probe/root, and greedy totals of 1.572, 0.770, and 0.818 seconds, with unchanged paired savings.

Direct DP favors small families; sharing becomes useful as exact solves grow costlier. Beyond matching retention, a 64-group cardinality/quota study uses 1,536 observations and up to 635,376

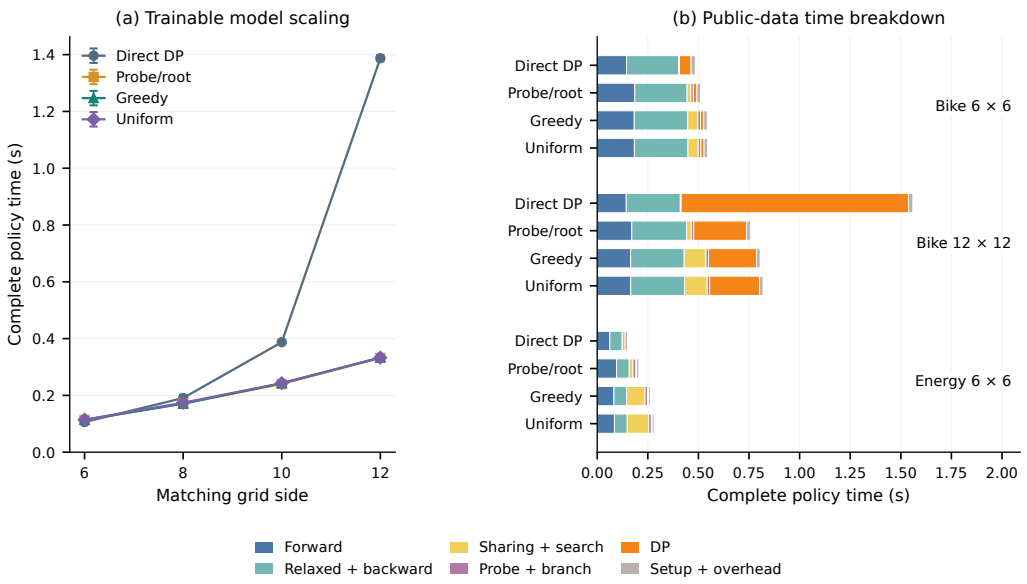


Figure 3: MLP runtime and its components. Synthetic grids vary from 6 to 12 with twelve observations per group; public-data costs include fixed-observation Bike comparisons. The cost decomposition charges forward-error calculations and exact fallback. Means use ten seeds after averaging tail masses; error bars indicate 95% intervals.

branches. The tree takes 3.5122 ms per endpoint versus 16.0200 ms for decision MILP (Section D.9). These endpoint-checking times complement the complete training comparisons. Group size, tail mass, and constraints determine exact-solver cost; shared bounds and tree expansion add overhead.

Warm-started full and decision MILP retain the size-dependent comparison (Section D.4). Always accepting relaxed proposals gives lower mean heldout MSE and worst-case CVaR in all three linear public-data settings, while allowing training-risk increases (Table 3). The network's four checking policies have identical heldout metrics.

5.5 When numerical bounds leave a decision unresolved

Wider risk intervals can leave more decisions unresolved (Equation (5)). On 140 fixed endpoints, raising the enclosure radius from 10^{-6} to 10^{-2} increases unknown comparisons from zero to 97 and mean queries from 1.42 to 4.86; every returned sign agrees with enumeration (Figure 8 and Table 20). In complete training, raising DP padding from 10^{-8} to 10^{-7} at $\gamma = 10^{-8}$ changes greedy's fallback rate by 0.032 ± 0.071 percentage points and time by 0.020 ± 0.034 seconds (Table 9). These intervals span zero. Table 19 reports the pre-fallback outcomes recovered from all saved network proposals (Figure 23); Section E.1 adds final outcomes, near-boundary comparisons, and range checks.

6 Discussion

Threshold sharing lowers the cost of enforcing empirical risk descent. Further search helps when avoided exact solves repay its cost (Figure 7).

Experiments use at most 1,728 risk-relevant observations and a width-16 network. Matching DP grows exponentially with grid size; numerous feasible subsets do not establish large-data scalability. Larger datasets and networks remain untested. The unchecked linear control has lower mean heldout risks, so verified training descent does not imply predictive improvement.

Reproducibility Statement

The appendix provides complete proofs, algorithm definitions, data preparation, study settings, and numerical error bounds. Supplementary tables report aggregate results and decision counts. The full implementation and detailed experimental records are available from the corresponding author upon reasonable request after anonymous review; review-time requests can use the venue's anonymous discussion channel (Section C.11).

AI Use Statement

In preparing the final version of this paper, large language models were used for language polishing and the analysis and summarization of experimental results. All other aspects of the work, including problem formulation, theoretical development, experimental design and implementation, scientific interpretation, and final manuscript decisions, are the responsibility of the human authors.

References

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/9ce3c52fc54362e22053399d3181c638-Abstract.html.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 136–145, 2017. URL https://proceedings.mlr.press/v70/amos17a.html.

Debraj Banerjee, Santanu Mahapatra, and Kunal Narayan Chaudhury. Local-minima-preserving polynomial relaxation of Ising problems. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=EBTMJ3MXt0.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9508–9519, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6bb56208f672af0dd65451f869fedfd9-Abstract.html.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5230–5242, 2022. doi: 10.52202/068431-0378. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/228b9279ecf9bbafe582406850c57115-Abstract-Conference.html.

Jérôme Bolte, Tam Le, Edouard Pauwels, and Tony Silveti-Falls. Nonsmooth implicit differentiation for machine-learning and optimization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13537–13549, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/70afbf2259b4449d8ae1429e054df1b1-Abstract.html.

Rudy Bunel, Jingyue Lu, Ilker Turkaslan, Philip H. S. Torr, Pushmeet Kohli, and M. Pawan Kumar. Branch and bound for piecewise linear neural network verification. *Journal of Machine Learning Research*, 21(42):1–39, 2020. URL https://jmlr.org/papers/v21/19-468.html.

Luis Candanedo. Appliances energy prediction. UCI Machine Learning Repository, 2017. URL https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction. Dataset, licensed under CC BY 4.0.

Luis M. Candanedo, Véronique Feldheim, and Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, 140:81–97, 2017. doi: 10.1016/j.enbuild.2017.01.083. URL https://www.sciencedirect.com/science/article/pii/S0378778816308970.

Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint. *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*. SIAM, 2022. doi: 10.1137/1.9781611976991. URL https://doi.org/10.1137/1.9781611976991.

Xiaoyu Chen, Marc Goerigk, and Michael Poss. The robust selection problem with information discovery. *Discrete Applied Mathematics*, 382:277–292, 2026. doi: 10.1016/j.dam.2025.12.012. URL https://arxiv.org/abs/2501.02510.

John M. Danskin. *The Theory of Max-Min and its Application to Weapons Allocation Problems*. Springer, 1967. doi: 10.1007/978-3-642-46092-0. URL https://link.springer.com/book/10.1007/978-3-642-46092-0.

Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146:37–75, 2014. doi: 10.1007/s10107-013-0677-5. URL https://doi.org/10.1007/s10107-013-0677-5.

Haya Diwan, Lisa Hellerstein, Nicole Megow, and Jens Schlöter. Optimal verification of a minimum-weight basis in an uncertainty matroid. In *43rd International Symposium on Theoretical Aspects of Computer Science*, volume 364 of *Leibniz International Proceedings in Informatics*, pp. 32:1–32:20, 2026. doi: 10.4230/LIPIcs.STACS.2026.32. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.STACS.2026.32.

Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, 1972. doi: 10.1145/321694.321699.

Nicolas El Maalouly. Exact matching: Algorithms and related problems. In *40th International Symposium on Theoretical Aspects of Computer Science (STACS 2023)*, volume 254 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 29:1–29:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi: 10.4230/LIPIcs.STACS.2023.29. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.STACS.2023.29.

Guy Even and Shakhar Smorodinsky. Hitting sets online and unique-max coloring. *Discrete Applied Mathematics*, 178:71–82, 2014. doi: 10.1016/j.dam.2014.06.019. URL https://www.sciencedirect.com/science/article/pii/S0166218X14002947.

Hadi Fanaee-T. Bike sharing. UCI Machine Learning Repository, 2013. URL https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset. Dataset, licensed under CC BY 4.0.

Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2:113–127, 2014. doi: 10.1007/s13748-013-0040-3. URL https://link.springer.com/article/10.1007/s13748-013-0040-3.

Tomás Feder, Rajeev Motwani, Rina Panigrahy, Chris Olston, and Jennifer Widom. Computing the median with uncertainty. *SIAM Journal on Computing*, 32(2):538–547, 2003. doi: 10.1137/S0097539701395668. URL https://doi.org/10.1137/S0097539701395668.

Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Mind the duality gap: Safer rules for the lasso. In *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 333–342, 2015. URL https://proceedings.mlr.press/v37/fercoq15.html.

Riccardo Grazzi, Massimiliano Pontil, and Saverio Salzo. Nonsmooth implicit differentiation: Deterministic and stochastic convergence rates. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16250–16274, 2024. URL https://proceedings.mlr.press/v235/grazzi24a.html.

Zhishuai Guo and Tianbao Yang. Communication-efficient federated group distributionally robust optimization. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0726. URL https://papers.nips.cc/paper_files/paper/2024/hash/29021b06afa4c648ee438584f7ef3e7e-Abstract-Conference.html.

Manoj Gupta, Yogish Sabharwal, and Sandeep Sen. The update complexity of selection and related problems. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 13 of *Leibniz International Proceedings in Informatics*, pp. 325–338, 2011. doi: 10.4230/LIPIcs.FSTTCS.2011.325. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.FSTTCS.2011.325.

András Gyárfás and Jenő Lehel. Covering and coloring problems for relatives of intervals. *Discrete Mathematics*, 55(2):167–180, 1985. doi: 10.1016/0012-365X(85)90045-7. URL https://users.renyi.hu/~gyarfas/Cikkek/23_GyarfasLehel_CoveringAndColoringProblemsForRelativesOfIntervals.pdf.

Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, second edition, 2002. doi: 10.1137/1.9780898718027. URL https://epubs.siam.org/doi/book/10.1137/1.9780898718027.

Cheng-Han Huang, Yongliang Sun, Chaoyan Huang, Ismail Alkhouri, and Rongrong Wang. Local minima in quadratic-penalty relaxations of binary linear programs. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=BPZunjhIol.

Yuma Ichikawa. Controlling continuous relaxation for combinatorial optimization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 47189–47216. Curran Associates, Inc., 2024. doi: 10.52202/079017-1495. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/54191f424e9013fc1d7b923f6e45dff4-Paper-Conference.pdf.

Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher (eds.), *Complexity of Computer Computations*, pp. 85–103. Plenum Press, 1972. doi: 10.1007/978-1-4684-2001-2_9. URL https://link.springer.com/chapter/10.1007/978-1-4684-2001-2_9.

Adam Kasperski and Paweł Zieliński. Robust recoverable and two-stage selection problems. *Discrete Applied Mathematics*, 233:52–64, 2017. doi: 10.1016/j.dam.2017.08.014. URL https://www.sciencedirect.com/science/article/pii/S0166218X17304006.

Wouter Kool, Herke van Hoof, and Max Welling. Stochastic beams and where to find them: The Gumbel-Top-k trick for sampling sequences without replacement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3499–3508. PMLR, 2019. URL https://proceedings.mlr.press/v97/kool19a.html.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955. doi: 10.1002/nav.3800020109.

A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960. doi: 10.2307/1910129. URL https://doi.org/10.2307/1910129.

Daniel Levy, Yair Carmon, John C. Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8847–8860, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/64986d86a17424eeac96b08a6d519059-Abstract.html.

Haifeng Li and Mo Hai. CASP: Learning-augmented offline approximation with verifiable certificates and bounded-loss PAC guarantees. arXiv:2607.14545, 2026. URL https://arxiv.org/abs/2607.14545.

Junyi Liu and Jong-Shi Pang. Risk-based robust statistical learning by stochastic difference-of-convex value-function optimization. *Operations Research*, 71(2):397–414, 2022. doi: 10.1287/opre.2021.2248. URL https://doi.org/10.1287/opre.2021.2248. Published online February 2022; March–April 2023 issue.

Chris J. Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://papers.nips.cc/paper_files/paper/2014/hash/937debc749f041eb5700df7211ac795c-Abstract.html.

Nicole Megow and Jens Schlöter. Set selection under explorable stochastic uncertainty via covering techniques. In *Integer Programming and Combinatorial Optimization*, volume 13904 of *Lecture Notes in Computer Science*, pp. 319–333. Springer, 2023. doi: 10.1007/978-3-031-32726-1_23. URL https://arxiv.org/abs/2211.01097.

Nabil H. Mustafa. Computing optimal epsilon-nets is as easy as finding an unhit set. In *46th International Colloquium on Automata, Languages, and Programming*, volume 132 of *Leibniz International Proceedings in Informatics*, pp. 87:1–87:12, 2019. doi: 10.4230/LIPIcs.ICALP.2019.87. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2019.87.

Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Gap safe screening rules for sparsity enforcing penalties. *Journal of Machine Learning Research*, 18(128):1–33, 2017. URL https://jmlr.org/papers/v18/16-577.html.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103: 127–152, 2005. doi: 10.1007/s10107-004-0552-5. URL https://doi.org/10.1007/s10107-004-0552-5.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, 2007. URL https://proceedings.neurips.cc/paper/2007/hash/013a006f03dbc5392effeb8f18fda755-Abstract.html.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000. doi: 10.21314/JOR.2000.038. URL https://doi.org/10.21314/JOR.2000.038.

R. Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002. doi: 10.1016/S0378-4266(02)00271-6. URL https://doi.org/10.1016/S0378-4266(02)00271-6.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958. doi: 10.2140/pjm.1958.8.171. URL https://msp.org/pjm/1958/8-1/p14.xhtml.

Tasuku Soma and Yuichi Yoshida. Statistical learning with conditional value at risk. arXiv:2002.05826, 2020. URL https://arxiv.org/abs/2002.05826.

Tasuku Soma, Khashayar Gatmiry, Sharut Gupta, and Stefanie Jegelka. Near-optimal algorithms for group distributionally robust optimization and beyond. arXiv:2212.13669v2, 2025. URL https://arxiv.org/abs/2212.13669v2. Revised version of a preprint first posted in 2022.

Philip Sosnin, Matthew Wicker, Josh Collyer, and Calvin Tsay. Abstract gradient training: A unified certification framework for data poisoning, unlearning, and differential privacy. *Journal of Machine Learning Research*, 27(128):1–58, 2026. URL https://www.jmlr.org/papers/v27/25-2206.html.

Guillaume Van Dessel and François Glineur. Optimal inexactness schedules for tunable oracle-based methods. *Optimization Methods and Software*, 39(3):664–698, 2024. doi: 10.1080/10556788.2023.2296982. URL https://www.tandfonline.com/doi/full/10.1080/10556788.2023.2296982.

Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkevoJSYPB.

John von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. In H. W. Kuhn and A. W. Tucker (eds.), *Contributions to the Theory of Games, Volume II*, number 28 in Annals of Mathematics Studies, pp. 5–12. Princeton University Press, 1953. doi: 10.1515/9781400881970-002.

Haoyu Peter Wang, Nan Wu, Hang Yang, Cong Hao, and Pan Li. Unsupervised learning for combinatorial optimization with principled objective relaxation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 31444–31458. Curran Associates, Inc., 2022. doi: 10.52202/068431-2280. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/cbc1ad2066f0afebbcea930c5688fc1f-Paper-Conference.pdf.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1658–1665, 2019. doi: 10.1609/aaai.v33i01.33011658. URL https://ojs.aaai.org/index.php/AAAI/article/view/3982.

Runtian Zhai, Chen Dan, J. Zico Kolter, and Pradeep Ravikumar. DORO: Distributional and outlier robust optimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12345–12355, 2021. URL https://proceedings.mlr.press/v139/zhai21a.html.

Shushang Zhu and Masao Fukushima. Worst-case conditional value-at-risk with application to robust portfolio management. *Operations Research*, 57(5):1155–1168, 2009. doi: 10.1287/opre.1080.0684. URL https://pubsonline.informs.org/doi/10.1287/opre.1080.0684.

A FULL PROOFS

The proofs follow the main argument: relaxed information can leave a proposed update ambiguous; shared thresholds can resolve it; query and tree bounds describe the work required. The numerical results then show how much accuracy is needed to preserve those decisions.

A.1 NOTATION AND THE DECISION INTERFACE

The tail mass β complements the usual CVaR confidence level: $\beta = 1/4$ averages the worst quarter of losses. We use $\mathbb{E}$ and $\Pr$ for expectation and probability, $|S|$ for set cardinality, $\|x\|$ for the Euclidean norm, and $\mathbf{1}_A$ for the indicator of event A. The symbols $\lfloor x \rfloor$ and $\lceil x \rceil$ round down and up to integers. Both $\log$ and $\ln$ denote natural logarithms unless a base is specified; the constant e inside these logarithms is their base. Hats mark computed approximations, while overbars mark quantities defined locally. A numerical enclosure is simply an interval known to contain the exact value.

Symbol	Meaning
$\theta, y = \theta + s$	Current and proposed model parameters
$B \in \mathcal{B}$	A permitted collection of retained groups (a branch)
$F_B(\theta, t), f_B(\theta)$	Variational branch objective and its threshold minimum
$V_{\mathrm{D}}, V_{\mathrm{H}}, \Gamma$	Discrete risk, relaxed risk, and gap $V_{\mathrm{H}} - V_{\mathrm{D}}$
$[L_0, U_0], [\ell_B, u_B]$	Intervals containing current risk and proposed branch risk
$[L_1, U_1]$	Bounds on the complete proposed-endpoint risk
$I_B(L_0)$	Thresholds satisfying $F_B(y, t) < L_0$
P	Cover size for the complete branch family
$P_{\mathcal{C}}$	Cover size after initial screening
P_ϵ	Candidate cover at the lower level $L_0 - \epsilon$
$\mathcal{C}, K$	Initially unresolved branches and their count
K_0	Number of branches unresolved at a retained threshold t_0
h	Maximum root-to-leaf depth of the binary retained-set tree
γ, μ, τ, ζ	Required decrease, available separation, requested accuracy, threshold suboptimality
d, η, L	Proposal direction, step size, and branch smoothness bound
G, s_g, k, m	Number of groups, group size, retained groups, retained observations
n	Side length of the matching grid (locally, a generic set size)
β	Tail mass, with $0 < \beta < 1$
Q_b, Q_s	Numbers of branch queries and threshold sweeps
g_H, t_H	Relaxed first-order direction and minimizing common threshold
c	Risk level to beat: L_0 or, with a margin, $L_0 - \gamma$

The full cover is defined when every branch interval is nonempty; discovery uses the screened cover $P_{\mathcal{C}}$, and numerical overestimate ϵ gives the effective cover P_ϵ; $P_{\mu/2}$ is its margin-based specialization. In the implicit tree, retaining t_0 from the initial pool gives $K \le K_0$, with tree work controlled by K_0. Common letters are reused only within explicitly stated local settings: δ is a gap scale in the two-group construction and a failure probability in the query bound; ϵ denotes either a numerical overestimate or a decision-error probability as stated. A representative is the threshold selected for a branch by a fixed rule; a canonical minimizer uses a specified deterministic rule among all minimizing thresholds. A witness is a feasible branch whose risk lower bound proves nondecrease. An L-smooth branch satisfies $\|\nabla f(u) - \nabla f(v)\| \le L\|u - v\|$ along the segment under consideration. Branch queries return an optimized endpoint value and threshold; sweeps evaluate all branches at a shared threshold. The lower-bound interface supplies initial information and charges these endpoint operations. Explicit-data implementations additionally charge losses and score aggregation.

A.2 COMPLETE PROOF OF THE TWO-GROUP CONSTRUCTION

The construction changes a loss that is ignored by the common relaxed threshold but still contributes to a branch risk. We first calculate the risk of two observations. For ordered losses $u < v$ and tail mass $3/4$, the function $F(t) = t + \frac{2}{3}\{(u - t)_+ + (v - t)_+\}$ has successive slopes $-1/3, 1/3, 1$. Its

unique minimizer is u, with

$$\min_t F(t) = \frac{1}{3}u + \frac{2}{3}v. \tag{11}$$

We use x as the scalar model parameter, $\lambda > 0$ as a loss scale, and $\sigma \in \{-1, 1\}$ to select the two instances. Retaining one of two groups, define

$$A_\sigma(x) = \{3\lambda/10 + (2 + 3\sigma)x,\ 3\lambda - x\}, \qquad B(x) = \{2\lambda - x,\ 2\lambda - x\}. \tag{12}$$

Set $\lambda = 30\delta/7$ and work on $\mathcal{U} = (-\lambda/40, \lambda/10)$. The low A observation lies below the candidate common threshold $t_H = \lambda - x$: for $\sigma = 1$, this follows from $6x < 3\lambda/5 < 7\lambda/10$; for $\sigma = -1$, the separation is $7\lambda/10$. Both high losses exceed t_H, fixing their order. By Equation (11),

$$f_{A,\sigma}(x) = 21\lambda/10 + \sigma x, \qquad f_B(x) = 2\lambda - x.$$

Their difference $\lambda/10 + (\sigma + 1)x$ exceeds $\lambda/20$ on $\mathcal{U}$, so $V_{\mathrm{D}}^\sigma = f_{A,\sigma}$ is smooth there. Near t_H, the active hinges give

$$F_{A,\sigma}(x, t) = \frac{1}{3}t + \frac{2}{3}(3\lambda - x), \qquad F_B(x, t) = -\frac{1}{3}t + \frac{4}{3}(2\lambda - x).$$

They intersect at t_H, with value $7\lambda/3 - x$. Their opposite threshold slopes make the convex maximum strictly decrease immediately before the intersection and increase immediately after it; hence this is the unique global minimizer. At zero the gap is $7\lambda/3 - 21\lambda/10 = 7\lambda/30 = \delta$. The partial derivatives are

$$\partial_x F_A = -2/3, \quad \partial_x F_B = -4/3, \qquad \partial_t F_A = 1/3, \quad \partial_t F_B = -1/3.$$

Equal minimax weights give relaxed derivative -1. These derivatives, the threshold, and active values agree between instances. Taking $d = 1$ and $\eta = \delta/4 = 7\lambda/120 \in \mathcal{U}$ yields $\Delta_{\mathrm{D}}^\sigma = \sigma\eta$. The differing low-loss derivatives 5 and -1 are hidden by an inactive relaxed hinge. Section A.6 also makes the initial raw derivatives agree.

A.3 Threshold intervals and their optimal cover

Proof of the variational characterization At endpoint y and level $c = L_0$, continuity and convexity make each nonempty $I_B(c) = \{t : F_B(y, t) < c\}$ an open interval; coercivity makes it bounded. Attainment gives $f_B(y) < c$ exactly when $I_B(c)$ is nonempty. Finiteness of $\mathcal{B}$ then gives part 1 of Theorem 2. Likewise, a common point satisfies $\max_B F_B(y, t) < c$, and exists exactly when $\min_t \max_B F_B(y, t) < c$, proving part 2. For finite T,

$$\max_B \min_{t \in T} F_B(y, t) < c \quad \Longleftrightarrow \quad \forall B\ \exists t \in T :\ F_B(y, t) < c.$$

The right side says that T intersects every sublevel interval.

Proof that piercing equals packing, with strict endpoints Let τ be the minimum piercing size and ν the maximum disjoint-family size. One point meets at most one disjoint interval, so $\tau \geq \nu$. For the converse, choose a remaining interval J with smallest right endpoint r. Every interval intersecting J starts strictly before r and ends at or after r. Finiteness permits a point between their largest left endpoint and r, covering them all. Remove the covered intervals and repeat. The selected witnesses J are pairwise disjoint, with one covering point each, so $\tau \leq \nu$. If all remaining right endpoints are $+\infty$, choose a point beyond their finite left endpoints. This proves the interval transversal identity (Gyárfás & Lehel, 1985), including strict endpoints and unbounded intervals.

Cover size and adaptive query count The cover $P(y, L_0)$ describes known endpoint functions; discovering its locations costs queries. Branch minimizers can lie outside a common intersection, so their sequential discovery can exceed the optimal cover (Section B.5). We therefore count branch queries and sweeps separately.

A.4 Sound decisions, branch screening, and numerical error

Theorem 5 (Soundness and a local query bound)**.** *Let all supplied bounds contain their corresponding values. Every decision of* BranchCheck *is sound. Use the initial candidate family*

and count in Equation (8). With a budget permitting K exact branch queries, the algorithm uses at most K branch queries and K additional sweeps. If $V_{\mathrm{D}}(y) < L_0$ or $V_{\mathrm{D}}(y) \geq U_0$, it reaches the corresponding decision. With query upper errors at most w, the descent conclusion is reached whenever $V_{\mathrm{D}}(y) + w < L_0$. With lower errors at most w, a counterexample is found whenever $V_{\mathrm{D}}(y) - w \geq U_0$.

The interval decision test For $V_{\mathrm{D}}(\theta) \in [L_0, U_0]$ and $f_B(y) \in [\ell_B, u_B]$, taking maxima gives

$$V_{\mathrm{D}}(y) \in [L_1, U_1], \qquad L_1 = \max_B \ell_B, \quad U_1 = \max_B u_B.$$

A valid relaxation upper bound $\overline{H}_y$ can reduce U_1 to $\min\{U_1, \overline{H}_y\}$. It follows that

$$\Delta_{\mathrm{D}} \in [L_1 - U_0, U_1 - L_0]. \tag{13}$$

A negative upper endpoint establishes descent. A nonnegative lower endpoint establishes nondecrease, witnessed by a branch attaining L_1. In the independent interval model, any baseline and branch vector in their product intervals is admissible, subject only to the optional hull cap. If this completion set is nonempty, its minimum and maximum endpoint values are $\max_B \ell_B$ and $\min\{\max_B u_B, \overline{H}_y\}$, both attainable. Thus both endpoints of Equation (13) are attainable: failure of either test leaves a completion with the opposite verdict. Thus no further sign decision follows from these independent intervals alone. Correlated bounds can first narrow the completion set.

Proof of soundness and the K bound Intersecting branch-query enclosures preserves validity. Sharing a feasible threshold also preserves it because $f_C(y) \leq F_C(y, t)$ for every C. Thus all upper bounds decrease monotonically and the interval test stays sound. Initially screened branches remain screened, and each eligible branch is queried at most once at a fixed precision, giving $Q_b \leq K$ and $Q_s \leq K$. An initial sweep raises the mixed-operation bound to $2K + 1$.

With exact queries and $V_{\mathrm{D}}(y) < L_0$, every query removes its branch, exhausting the candidate set within K queries. If $V_{\mathrm{D}}(y) \geq U_0$, a maximizing branch's valid upper bound stays at least U_0; it must be queried before exhaustion and supplies a counterexample. Values between L_0 and U_0 can remain unknown because the baseline is uncertain. With upper errors at most w, $V_{\mathrm{D}}(y) + w < L_0$ again ensures self-removal; with lower errors at most w, $V_{\mathrm{D}}(y) - w \geq U_0$ ensures a maximizing branch witnesses nondecrease. A feasible approximate threshold remains a sound upper bound, with its residual controlling informativeness.

How the step size determines K. A branch far below the current risk need not be queried if its possible increase is too small to matter. Suppose $a_B \geq f_B(\theta)$ is a stored upper bound, g_B approximates its gradient with norm error at most e_B, and L_B bounds the gradient Lipschitz constant along the proposed step s. The linear prediction plus gradient and curvature allowances gives

$$u_B^0 = a_B + g_B^\top s + e_B\|s\| + \tfrac{1}{2}L_B\|s\|^2. \tag{14}$$

Writing $\alpha_B = L_0 - a_B$, candidate membership becomes $\alpha_B \leq g_B^\top s + e_B\|s\| + L_B\|s\|^2/2$. This is a directional stability condition: only branches whose possible movement bridges their current deficit need further work. The result is conditional on the stated cache; its acquisition is charged. For nonsmooth CVaR branches, feasible threshold evaluations supply bounds directly, without requiring Equation (14) across a tail-order switch.

Derivation of the directional screen For a differentiable branch with L_B-Lipschitz gradient along the segment,

$$\begin{aligned} f_B(\theta + s) - f_B(\theta) &= \int_0^1 \nabla f_B(\theta + as)^\top s \, da \\ &\leq \nabla f_B(\theta)^\top s + \int_0^1 L_B a\|s\|^2 \, da = \nabla f_B(\theta)^\top s + \tfrac{1}{2}L_B\|s\|^2. \end{aligned}$$

Substitute $f_B(\theta) \leq a_B$ and $\nabla f_B(\theta)^\top s \leq g_B^\top s + e_B\|s\|$ to obtain Equation (14), including initial value and gradient uncertainty. Comparison with L_0 screens branches whose bounded movement cannot close their current deficit. The threshold screen in Equation (7) also handles active-tail changes without branch smoothness.

A finite precision schedule If numerical error prevents a decision, progressively halve the allowed error. At refinement level $j \geq 0$, start from allowance $w_0 > 0$ and request upper and lower errors $w_j = w_0 2^{-j}$, retaining valid bounds and querying only still-relevant branches among the original K. For a fixed baseline and $\mu = \max\{L_0 - V_{\mathrm{D}}(y), V_{\mathrm{D}}(y) - U_0\} > 0$, Theorem 5 decides once $w_j < \mu$. It suffices to refine through level

$$j_* = \max\{0, \lceil \log_2(2w_0/\mu) \rceil\}$$

for at most $K(1 + j_*)$ branch queries and as many additional sweeps. Solver work can be charged per requested accuracy. An obstructing baseline width also requires refinement, implemented by a charged exact refresh.

A.5 Adaptive discovery of a threshold cover

This proof measures how much a uniformly selected branch can remove from the unresolved family. Write $F_i(t) = F_i(y, t)$ and $c = L_0$. We prove Theorem 3 for any fixed representative t_i in the nonempty interval $I_i = \{t : F_i(t) < c\}$. Canonical minimizers are one such choice. Start with $K \geq 1$ unresolved branches $\mathcal{C}$; $K = 0$ already establishes descent. Each iteration selects a current identifier uniformly, queries its representative, and removes all intervals containing it. Additional sound removals are allowed, and initially screened branches remain below c by their valid upper bounds.

Lemma 6 (Rank removal, including ties)**.** *Assign each candidate interval to a point p of an optimal piercing set, and split each assigned group into $t_B \leq p$ and $t_B > p$. Querying rank r in increasing order in a left class, or decreasing order in a right class, removes at least its first r remaining members. Ties use any fixed identifier order.*

Proof. In a left class, $t_C \leq t_B \leq p$ and $t_C, p \in I_C$ imply $t_B \in I_C$ by convexity. Reverse the inequalities on the right. Equal representatives cover the tied block and its preceding identifiers. □

A fixed partition used only for analysis Fix an optimal piercing set $\{p_1, \ldots, p_P\}$, $P = P_{\mathcal{C}}$. Assign each interval to one contained point by a fixed rule, and split each assigned group into

$$C_{k,-} = \{i : t_i \leq p_k\}, \qquad C_{k,+} = \{i : t_i > p_k\}.$$

There are $J \leq \min\{2P, K\}$ nonempty fixed classes, used only in the analysis. In a left class, $t_j \leq t_i \leq p_k$ and $t_j, p_k \in I_j$ imply $t_i \in I_j$; thus querying i removes every representative at most t_i. Right classes have the reversed ordering. Interior membership makes the argument valid for strict open sublevels.

Conditional on the history and selecting a class of size n, its selected identifier is uniform. Order representatives increasingly on the left and decreasingly on the right, with fixed identifier tie-breaking. The uniform rank $r \in \{1, \ldots, n\}$ removes at least its first r members; ties and cross-class deletions can only increase removals.

Expected progress per query To count the remaining work, assign a class of size n the harmonic weight $H_n = \sum_{q=1}^{n} q^{-1}$, with $H_0 = 0$. For remaining class sizes n_j, the total weight is $\Phi = \sum_j H_{n_j}$. To compute the expected remaining weight, sum the harmonic weights of all possible remaining class sizes:

$$\sum_{q=0}^{n-1} H_q = \sum_{s=1}^{n-1} \frac{n-s}{s} = nH_{n-1} - (n-1) = n(H_n - 1)$$

A uniformly selected rank therefore gives, conditional on choosing a class of size n,

$$\mathbb{E}[H_{\text{remaining size}} \mid \text{class and history}] \leq \frac{1}{n} \sum_{r=1}^{n} H_{n-r} = H_n - 1.$$

Other class weights cannot increase, and each query removes its own branch, giving $Q_b \leq K$. Let $\mathcal{F}_q$ contain the information from the first q queries and Φ_q be the total remaining harmonic weight,

set to zero after termination. The expected decrease gives

$$\mathbb{E}[\Phi_q - \Phi_{q+1} \mid \mathcal{F}_q] \geq \mathbf{1}_{\{Q_b > q\}}, \qquad \Phi_0 \geq \sum_{q=0}^{K-1} \Pr(Q_b > q) = \mathbb{E}Q_b.$$

The second inequality telescopes over $q = 0, \ldots, K-1$. Since $\sum_j n_j^0 = K$ and $H_n \leq 1 + \ln n$, concavity gives

$$\Phi_0 \leq J\,[1 + \ln(K/J)]\,.$$

Because $x[1 + \ln(K/x)]$ increases on $(0, K]$, for $2P \leq K$ substitute $J \leq 2P$ and enlarge $\ln(K/(2P))$ to $\ln(K/P)$. For $2P > K$, use $\Phi_0 \leq K \leq 2P[1 + \ln(K/P)]$. Together with $Q_b \leq K$ this proves Equation (9); at most one sweep per query gives $Q_b + Q_s \leq 2Q_b$ after initialization. Empirical CVaR uses a deterministic tail quantile, including flat minima. Independent branch-routine randomness can also be presampled and conditioned on to obtain fixed representatives.

A quantitative probability bound For a bound on unusually long runs, count how many successive halvings would empty each class. Let $b(n)$ be that count and B_* its initial total, where n_j^0 is class j's initial size:

$$b(0) = 0, \qquad b(n) = \lfloor \log_2 n \rfloor + 1 \quad (n \geq 1), \qquad B_* = \sum_{j=1}^{J} b(n_j^0).$$

Since $b(n) \leq (1 + \ln n)/\ln 2$ for $n \geq 1$, the preceding class bound gives

$$B_* \leq \frac{J}{\ln 2}[1 + \ln(K/J)] \leq \frac{2P}{\ln 2}[1 + \ln(K/P)].$$

Call a step successful if it removes at least half its selected class. Uniform rank gives conditional success probability at least $1/2$. A success reduces n to at most $\lfloor n/2 \rfloor$, decreasing $b(n)$ by at least one. Thus the integer potential $\Psi = \sum_j b(n_j)$ starts at B_* and is positive while queries remain.

Let Z_q equal one when query q removes at least half of its selected class and zero otherwise. After termination, extend this sequence with independent fair coin flips. Then $\mathbb{E}[2^{-Z_q} \mid \mathcal{F}_{q-1}] \leq 3/4$, and repeated conditioning gives $\mathbb{E}[2^{-S_m}] \leq (3/4)^m$ for $S_m = \sum_{q=1}^{m} Z_q$. The event $Q_b > m$ implies $S_m < B_*$. Markov's inequality yields

$$\Pr(Q_b > m) \leq 2^{B_*}(3/4)^m. \tag{15}$$

Taking $m = \lceil 4(B_* + \ln(1/\delta)) \rceil$ makes this at most δ, since $4\ln(4/3) > \max\{1, \ln 2\}$. The deterministic cap and $B_* = O(P \log(eK/P))$ prove the high-probability statement while allowing dependent query outcomes.

Numerical precision and the effective cover Numerical upper bounds may lie above the true responses. To retain the same removal argument, lower the comparison level by their maximum overestimate ϵ. In the following statement, $U_i(t)$ is a computed upper bound, ζ bounds threshold suboptimality, and μ is the available separation between the largest branch risk and level c.

Corollary 7 (Cover discovery with numerical upper bounds)**.** *Suppose shared responses satisfy*

$$F_i(t) \leq U_i(t) \leq F_i(t) + \epsilon$$

at every queried threshold. Let $J_i = \{t : F_i(t) < c - \epsilon\}$ for $i \in \mathcal{C}$, and let P_ϵ be their piercing number. If every fixed representative belongs to its J_i, uniform selection and removals based on $U_i(t) < c$ satisfy

$$\mathbb{E}Q_b \leq \min\{K, 2P_\epsilon[1 + \ln(K/P_\epsilon)]\}.$$

The probability bound Equation (15) applies to the corresponding effective classes. A sufficient solver condition is

$$F_i(t_i) \leq f_i + \zeta, \qquad \max_{i \in \mathcal{C}} f_i \leq c - \mu, \qquad \mu > \epsilon + \zeta.$$

Proof. The margin puts t_i in J_i. Whenever $t \in J_j$, $U_j(t) \leq F_j(t) + \epsilon < c$, so numerical screening includes all effective-interval deletions used above. Extra deletions decrease the potential, and the enclosing upper values establish descent at level c. □

An absolute-error-e estimate plus e overestimates by at most $2e$, so use $\epsilon = 2e$. Exact canonical minimizers have $\zeta = 0$.

The stricter level can require more thresholds even when the original intervals share one point. For the nested example in Section B.5, changing level C to $C - \epsilon$ changes

$$I_i = (C - i - 1, C) \quad \text{to} \quad J_i = (C - i - 1 + \epsilon, C - i\epsilon), \qquad 0 < \epsilon < 1.$$

At $\epsilon = 1/4$, indices $i = 1, 8, 8^2, \ldots$ give disjoint effective intervals while all original intervals intersect. Thus numerical margin can increase the required cover; precision refinement restores the representative condition when the available budget permits.

A.6 A SMOOTH CVaR LOWER BOUND WITH SHARED THRESHOLD ACCESS

We construct Theorem 4's family. A common constant shift of losses, values, and thresholds ensures nonnegative losses while preserving every finite-step difference.

Definition of the family We need M candidate groups whose endpoint changes can be hidden independently, while the relaxed objective remains the same. Here $L > 0$ bounds branch curvature and $\eta > 0$ is the step length. We set κ as the curvature scale, v as the resulting endpoint value scale, a as the initial candidate deficit, and r as the magnitude of the initial slope. The shared quadratic shift h controls the relaxed decrease; Q separates the two fixed groups from the candidates. Choose

$$\kappa = \frac{L}{6}, \quad v = \kappa\eta^2, \quad a = \frac{v}{8}, \quad r = \frac{21\kappa\eta}{8}, \quad h(x) = -rx + \frac{5}{2}\kappa x^2, \quad Q = 40(M+1)v. \tag{16}$$

To keep different candidates visible on disjoint threshold intervals, space their low-loss locations p_i by $8v$. Choose the high-loss constants c_i so that all candidate risks start at $-a$: for $i = 1, \ldots, M$, let

$$p_i = -Q + 8iv, \qquad c_i = -\frac{5}{2}a - \frac{3}{2}p_i - v.$$

Index instances by hidden bits $b \in \{0,1\}^M$. Retain one of $M + 2$ groups, each with three observations, and set $\beta = 5/6$. Two fixed groups determine the relaxed optimum independently of the bits:

$$A(x) = \{-Q + h(x), -Q + h(x), 3Q/2 + h(x)\}, \quad B(x) = \{h(x), h(x), h(x)\}.$$

Candidate group i redistributes two losses by opposite quadratic amounts. Their sum stays fixed while their tail-weighted average can change:

$$C_i^b(x) = \left\{ p_i + h(x) - \frac{5}{2}b_i\kappa x^2, \quad p_i + v + h(x) + \frac{5}{2}b_i\kappa x^2, \quad c_i + h(x) \right\}. \tag{17}$$

All claims below hold for $0 \le x \le \eta$.

Loss ordering and smoothness Writing u_i, m_i, w_i for the entries of Equation (17), we have $m_i - u_i = v + 5b_i\kappa x^2 > 0$ and, since $p_i \le -(32M + 40)v$,

$$w_i - m_i = -\tfrac{5}{2}a - \tfrac{5}{2}p_i - 2v - \tfrac{5}{2}b_i\kappa x^2 \ge -\tfrac{5}{2}p_i - \tfrac{77}{16}v > 0.$$

Their order is fixed. Tail mass $5/6$ assigns weights $1/5, 2/5, 2/5$, so

$$f_i^b(x) = \tfrac{1}{5}u_i + \tfrac{2}{5}m_i + \tfrac{2}{5}w_i = -a + h(x) + \tfrac{1}{2}b_i\kappa x^2, \tag{18}$$

$$f_A(x) = f_B(x) = h(x). \tag{19}$$

The fixed-group constants cancel with total low weight $3/5$ and high weight $2/5$. Candidate loss curvatures are $(5 - 5b_i)\kappa$, $(5 + 5b_i)\kappa$, and 5κ, all nonnegative. Branch curvatures are $(5 + b_i)\kappa \le 6\kappa = L$, with fixed-group curvature 5κ. Thus the losses are convex and smooth and the branches convex and L-smooth.

The common relaxed optimum For this calculation, use the shifted threshold coordinate $\zeta = t - h(x)$, giving $F(x,t) = h(x) + \overline{F}(x,\zeta)$. For $-Q < \zeta < 0$, the fixed groups satisfy

$$\overline{F}_A(\zeta) = \frac{3}{5}\zeta + \frac{3}{5}Q, \qquad \overline{F}_B(\zeta) = -\frac{1}{5}\zeta.$$

Their opposite slopes give a unique minimizer at $\zeta_H = -3Q/4$, value $3Q/20$, and minimax weights $1/4, 3/4$. Every candidate middle loss is below this threshold because

$$\zeta_H - \{p_i + v + \tfrac{5}{2}b_i\kappa x^2\} \geq Q/4 - 8Mv - 7v/2 = (2M + 13/2)v > 0.$$

Its high loss is above it since $c_i > 0 > \zeta_H$. Only that hinge contributes:

$$\overline{F}_i(x, \zeta_H) = \tfrac{3}{5}\zeta_H + \tfrac{2}{5}c_i = \frac{3Q}{20} - \left(\frac{24i}{5} + \frac{21}{40}\right)v < \frac{3Q}{20}.$$

The anchors lower-bound the maximum everywhere, and all candidates lie below them at their minimizer. Therefore

$$V_{\mathrm{H}}^b(x) = \frac{3Q}{20} + h(x), \qquad t_H(x) = -\frac{3Q}{4} + h(x), \qquad \frac{d}{dx}V_{\mathrm{H}}^b(x) = h'(x). \tag{20}$$

These functions and all active and inactive responses at t_H are bit-independent.

Initial information, step signs, and the candidate count Bits enter through x^2, so initial observation values are shared, and all observation and branch derivatives equal $-r$. Initial branch values are $-a$ for candidates and zero for the fixed groups, giving $V_{\mathrm{D}}^b(0) = 0$ and gap $3Q/20$. At the endpoint, $h(\eta) = -21v/8 + 5v/2 = -v/8$. Candidate values are $-v/4$ for bit zero and $v/4$ for bit one. Hence

$$\Delta_{\mathrm{D}} = \begin{cases} -v/8, & b = 0, \\ v/4, & b \neq 0. \end{cases} \qquad V_{\mathrm{H}}^b(\eta) - V_{\mathrm{H}}^b(0) = -v/8.$$

Since the initial relaxed derivative is $-r < 0$, the displacement η follows its descent direction (learning rate η/r for the unnormalized negative gradient). Public candidate curvature L gives Taylor bound

$$-a - r\eta + \tfrac{1}{2}L\eta^2 = -\tfrac{1}{8}v - \tfrac{21}{8}v + 3v = v/4.$$

Known fixed-group curvature 5κ gives exact upper value $-v/8$, leaving exactly $K = M$ candidates. The initial gap is

$$\Gamma(0) = 3Q/20 = 6(M+1)v = (M+1)L\eta^2. \tag{21}$$

Any prescribed gap $\delta > 0$ follows by setting $\eta = \sqrt{\delta/((M+1)L)}$, with endpoint margins $\delta/[48(M+1)]$ and $\delta/[24(M+1)]$. For nonnegative losses, add $Q + 4v$: the segment minimum $\min h = -441v/640 > -v$ bounds the fixed groups below by $-Q - v$ and candidate lows by $-Q + 8v - v - 5v/2 > -Q$.

Disjoint intervals that reveal individual bits At the endpoint, write $s_* = h(\eta) = -v/8$. After removing this shift, candidate i has losses $p_i - 5b_iv/2$, $p_i + v + 5b_iv/2$, and c_i. Below $p_i - 5v/2$, both changed hinges enter with preserved sum $2p_i + v$; above $p_i + 7v/2$, neither enters. The third is unchanged. Thus the response depends on b_i only inside

$$J_i = s_* + [p_i - 5v/2,\ p_i + 7v/2]. \tag{22}$$

Responses also agree at the endpoints. Width $6v$ and spacing $8v$ leave pairwise gaps $2v$, so any sweep exposes at most one bit. A branch solve exposes only its own bit, even if it returns endpoint observations or its full threshold function. Its minimizer is the low observation inside J_i, so sharing that threshold exposes no further bit. Minimizers of the fixed groups lie outside all private intervals.

Deterministic and randomized query lower bounds Supply the family form and constants, initial observations and first derivatives, initial branch values and gradients, curvature bounds, exact baseline, and complete relaxed solution and gradient trajectory. Access bit coefficients and endpoint observations through the two query types.

A deterministic all-zero run with fewer than M operations leaves a private region i unexamined. Flipping only b_i preserves its initial data and entire transcript but reverses the true sign, proving the

lower bound. For a randomized algorithm, couple all-zero and singleton e_i runs with shared coins. Let e_i be the bit vector with a single one at position i. Let E_i denote examination of region i on the all-zero run, and A_0, A_i denote descent outputs on the two runs. Outside E_i, transcripts and stopping decisions agree, so

$$|\Pr(A_0) - \Pr(A_i)| \le \Pr(E_i).$$

Instancewise error ϵ gives $\Pr(A_0) \ge 1 - \epsilon$ and $\Pr(A_i) \le \epsilon$, hence $\Pr(E_i) \ge 1 - 2\epsilon$. Since each operation examines at most one region,

$$Q_b + Q_s \ge \sum_{i=1}^{M} \mathbf{1}_{E_i}.$$

Expectations prove Equation (10); fixed positive costs c_b, c_s for a branch query and sweep, respectively give cost at least $\min(c_b, c_s)(1 - 2\epsilon)M$. This matches the upper bound's linear dependence on $K = M$.

Exact cover size of the hard family For the all-zero endpoint at level $V_{\mathrm{D}}(0) = 0$, candidate minima $-v/4$ give

$$I_i(0) = \left(s_* + p_i - \frac{5}{4}v,\ s_* + p_i + \frac{13}{12}v \right). \tag{23}$$

Below the low observation, slope $-1/5$ raises the minimum to zero after moving $5v/4$ left. Slope $1/5$ between the first two observations gives middle value $-v/20$; subsequent slope $3/5$ reaches zero another $v/12$ to the right. These determine both endpoints.

The fixed-group slopes are $(-1/5, 3/5)$ for A and $(-1/5, 1)$ for B, giving intervals

$$I_A(0) = \left(s_* - Q - \frac{5}{8}v,\ s_* - Q + \frac{5}{24}v \right), \quad I_B(0) = \left(s_* - \frac{5}{8}v,\ s_* + \frac{1}{8}v \right).$$

Candidate widths $7v/3$ and spacing $8v$ make them disjoint; the fixed-group intervals lie on either side. Thus the full cover has size $M + 2$, and initial smoothness screening leaves candidate cover M. A one-bit branch instead has value $v/4 > 0$ and an empty interval.

A.7 Accuracy, finite-step curvature, and oracle work

We ask how accurately endpoint responses must be evaluated to distinguish a decrease from an increase. The accuracy τ bounds both value error and the returned threshold's suboptimality. A call costs τ^{-p}, where $p > 0$ describes how computation grows as accuracy improves. We use the hard family of Section A.6, with M candidate groups and value scale $v = L\eta^2/6$.

Corollary 8 (The resolution required by a finite step)**.** *Consider inexact queries returning value estimates with error at most τ and, for a branch solve, a fixed threshold with residual at most τ. On the hard family, admissible responses can keep every sign instance indistinguishable whenever $\tau \ge v/2 = L\eta^2/12$. Any uniformly correct sign algorithm with error at most $\epsilon < 1/2$ therefore requires at least $(1-2\epsilon)M$ calls with $\tau < v/2$ in expectation. If a call costs τ^{-p}, $p > 0$, its expected additional work is at least $(1 - 2\epsilon)M(2/v)^p$.*

The inexact information interface At accuracy $\tau > 0$, a branch query returns $(\widehat{f}_i, t_i^\tau)$ satisfying

$$|\widehat{f}_i - f_i| \le \tau, \qquad F_i(t_i^\tau) - f_i \le \tau.$$

The threshold is fixed by the instance, identifier, and accuracy. A sweep returns $|\widehat{F}_i(t) - F_i(t)| \le \tau$ for every branch, giving valid upper bounds $\widehat{f}_i + \tau$ and $\widehat{F}_i(t) + \tau$. Uniform correctness requires a terminating correct binary answer with probability at least $1 - \epsilon$ for every instance and admissible response rule; unknown and nontermination count as failures. Following tunable-oracle work models (Van Dessel & Glineur, 2024), charge each query or sweep τ^{-p} for fixed $p > 0$, with initialization supplied separately.

Why coarse responses cannot distinguish the instances In Section A.6's hard family, the endpoint has $h(\eta) = -v/8$, $c = 0$, and $f_i^0 = -v/4$, $f_i^1 = v/4$. The common nonnegative-loss shift preserves the argument. Write $u_i^0 = p_i + h(\eta)$ and $z = t - u_i^0$. Evaluating the two changed hinges gives

$$F_i^1(t) - F_i^0(t) = \begin{cases} 0, & z \le -5v/2, \\ \frac{2}{5}(z + 5v/2), & -5v/2 \le z \le 0, \\ v, & 0 \le z \le v, \\ \frac{2}{5}(7v/2 - z), & v \le z \le 7v/2, \\ 0, & z \ge 7v/2. \end{cases} \tag{24}$$

The difference lies in $[0, v]$ and is supported on the disjoint private interval J_i. For coarse accuracy $\tau \ge v/2$, return $\widehat{F}_i(t) = (F_i^0(t) + F_i^1(t))/2$ in a sweep and exact responses for the fixed groups. These are bit-independent with error at most $v/2$. For a branch query, return

$$\widehat{f}_i = v/4, \qquad t_i^\tau = u_i^0.$$

For bit zero, u_i^0 minimizes the branch and the estimate error is $v/2$. For bit one, the estimate is exact, the minimizer is $u_i^0 - 5v/2$, and the intervening slope $1/5$ gives

$$F_i^1(u_i^0) - f_i^1 = \tfrac{1}{5}(5v/2) = v/2.$$

Both requirements therefore hold. Moreover, $F_i^0(u_i^0) = -v/4$ and $F_i^1(u_i^0) = 3v/4$, so the estimate is also within τ of the returned point's objective. For $\tau < v/2$, return exact values and canonical minimizers. These rules are history-independent, and every coarse answer is common to all bit instances, including adaptively chosen thresholds.

The number and cost of sufficiently accurate queries Call a query *fine* if $\tau < v/2$. Each fine solve accesses its own bit and each fine sweep at most one bit, since the J_i are disjoint; coarse calls access none. Apply the coupling proved in Section A.6 to the event E_i of fine access to bit i on the all-zero run. Until E_i, its transcript agrees with the singleton alternative under shared random coins, so $\Pr(E_i) \ge 1 - 2\epsilon$. Summing these events and charging each fine call at least $(2/v)^p$ gives

$$\mathbb{E}Q_{\text{fine}} \ge \sum_i \Pr(E_i) \ge (1 - 2\epsilon)M, \qquad \mathbb{E}W \ge (1 - 2\epsilon)M(2/v)^p.$$

This admissible oracle proves Corollary 8. An exact minimizing threshold at coarse accuracy would itself reveal its candidate bit.

A corresponding work upper bound To match the lower bound, choose an accuracy proportional to the available risk separation. Suppose $\max_i f_i \le c - \mu$ with $\mu > 0$ and $K \ge 1$; $K = 0$ needs no queries. Choose $\tau = \mu/4$. Sweep upper bounds overestimate by at most $\mu/2$, while branch representatives satisfy

$$F_i(t_i^\tau) \le f_i + \tau \le c - 3\mu/4 < c - \mu/2.$$

Let $P_{\mu/2}$ pierce these effective intervals $J_i = \{t : F_i(t) < c - \mu/2\}$ over the K initial candidates. Corollary 7, with at most one sweep per branch query, yields

$$\mathbb{E}W \le 2(4/\mu)^p \min\{K, 2P_{\mu/2}[1 + \ln(K/P_{\mu/2})]\}.$$

On the all-zero hard instance, $\mu = v/8$. Lowering its disjoint candidate intervals to level $-\mu/2$ preserves nonemptiness because their minima are $-v/4$. Hence $P_{\mu/2} = K = M$, and both work bounds have order $Mv^{-p} = M(L\eta^2)^{-p}$ for fixed p.

A.8 Implicit uniform discovery under adaptive threshold sharing

The flat procedure chooses uniformly from unresolved branches, but listing that family can be expensive. This section constructs the same query sequence in distribution using a tree, then bounds the extra node work.

Fix an endpoint and a nonempty feasible family $\mathcal{B}$ represented once by the leaves of a fixed binary tree of depth at most h. A leaf is one complete retained collection, and an internal node groups its feasible descendants. Each node's descendant count $n_N = |\mathcal{B}_N|$ is available without enumeration;

empty children are omitted. Let F_B be continuous and convex with attained minimum f_B, and fix comparison level c. Initialize a nonempty threshold pool T_0 independently of the new sampling randomness, retaining one $t_0 \in T_0$ throughout. For $T \supseteq T_0$, let $u_B(T)$ be the best shared bound on a branch and $U_N(T)$ the best common-threshold bound on a node:

$$u_B(T) = \min_{t \in T} F_B(t), \qquad U_N(T) = \min_{t \in T} \max_{B \in \mathcal{B}_N} F_B(t).$$

The node bound satisfies

$$\max_{B \in \mathcal{B}_N} u_B(T) \le U_N(T) \le \max_{B \in \mathcal{B}_N} F_B(t_0). \tag{25}$$

These bounds decrease as thresholds are added and agree at leaves. Discard nodes with $U_N(T) < c$. Define the initial candidate and families unresolved by the single retained threshold by

$$\mathcal{C} = \{B : u_B(T_0) \ge c\}, \qquad K = |\mathcal{C}|, \qquad \mathcal{C}_0 = \{B : F_B(t_0) \ge c\}, \qquad K_0 = |\mathcal{C}_0|.$$

Thus $\mathcal{C} \subseteq \mathcal{C}_0$ and $K \le K_0$; neither family needs to be enumerated.

Theorem 9 (An implicit implementation of uniform discovery)**.** *Charge the fixed probe separately and condition on its augmented pool T_0; counts below concern subsequent discovery. Use exact node bounds on a fixed binary tree of maximum depth h, with known feasible-leaf counts, updating each selected bound before acting. For any retained threshold $t_0 \in T_0$, let K count branches unresolved by the full set and K_0 those unresolved by t_0 alone:*

$$K = |\{B : \min_{t \in T_0} F_B(y,t) \ge L_0\}|, \qquad K_0 = |\{B : F_B(y,t_0) \ge L_0\}|.$$

If $V_{\mathrm{D}}(y) < L_0$ and $K \ge 1$, exact fixed minimizing thresholds and an unlimited budget give the same branch-query distribution as uniform flat discovery, hence Equation (9) and its probability bound. If $K = 0$, no branch is queried. At most hK_0 internal nodes are expanded and $1 + 2hK_0$ nodes generated.

Proof sketch. Random priorities order leaves uniformly. Pruning preserves that order, and every expanded node has an initially unresolved descendant. Counting its ancestors bounds node work (Section A.8). □

Cardinality trees have $h \le G$; balanced matching trees have $h \le \sum_{r=1}^{n} \lceil \log_2 r \rceil$. Consistent numerical envelopes preserve the coupling (Section A.8); a node budget stops a prefix.

Generating random priorities only when needed Conceptually assign independent leaf priorities $E_B \sim \mathrm{Exp}(1)$ and node minima $m_N = \min_{B \in \mathcal{B}_N} E_B$. Smaller priorities are processed first, so a node's minimum is the first of its leaf priorities. Since $-\log E_B$ is standard Gumbel, this is the exponential form of top-down Gumbel sampling (Maddison et al., 2014; Kool et al., 2019). Only needed minima are generated. Sample $m_{\mathrm{root}} \sim \mathrm{Exp}(n_{\mathrm{root}})$, where the parameter is the *rate*. To split a node with $m_N = t$, choose a winning nonempty child J with probability n_{N_J}/n_N and set $m_{N_J} = t$. Independently for the other children, sample

$$m_{N_j} = t + Z_j, \qquad Z_j \sim \mathrm{Exp}(n_{N_j}).$$

A sole nonempty child inherits its parent's minimum.

Lemma 10 (Conditional exponential split)**.** *This top-down procedure has exactly the joint law of the node minima induced by independent $\mathrm{Exp}(1)$ leaf priorities. It remains valid when the decision to expand a node depends on previously generated minima and previously obtained bounds.*

Proof. The exponential distribution with rate $a > 0$ has density $a\exp(-at)$ for $t \ge 0$ and mean $1/a$. The disjoint child sets have independent minima $M_j \sim \mathrm{Exp}(n_{N_j})$. Put $n = \sum_j n_{N_j}$, $M = \min_j M_j$, and $J = \arg\min_j M_j$. For $x_k > 0$, the joint density of $M = t$, $J = j$, and $M_k - t = x_k$ $(k \ne j)$ is

$$n_{N_j} e^{-n_{N_j} t} \prod_{k \ne j} n_{N_k} e^{-n_{N_k}(t + x_k)} = n_{N_j} e^{-nt} \prod_{k \ne j} n_{N_k} e^{-n_{N_k} x_k}.$$

Dividing by M's density ne^{-nt} gives the stated winner probabilities and independent losing increments. Conditional on each child's generated minimum, apply the same rule recursively inside that child. This constructs the joint law of any finite set of revealed minima; the unrevealed subtrees can be completed by the same conditional laws. Adaptive expansion chooses which conditional distribution to reveal next, and pruning leaves an unrevealed subtree in this complete coupling. Both preserve the law. □

The implicit search Start a priority queue with the root, keyed by persistent minima m_N. Pop the smallest key and recompute $U_N(T)$: discard if it is below c; otherwise split an internal node by Lemma 10, or query an unresolved leaf's value and fixed canonical minimizing threshold t_B, adding t_B to T. With exact baseline c, $f_B \geq c$ supplies a counterexample, while $f_B < c$ removes the branch by its own threshold. With an enclosed baseline, use $c = L_0$ for descent and a valid lower bound at least U_0 for a counterexample. Threshold additions update bounds, while keys remain fixed; the current bound is checked before every expansion or leaf query.

Lemma 11 (Equality of the queried-leaf sequences)**.** *For every fixed collection of distinct leaf priorities, the implicit search queries exactly the same sequence as the following virtual flat procedure: among the unqueried leaves with $u_B(T) \geq c$, choose the smallest priority, query its canonical threshold, update T, and repeat. This statement holds until either procedure stops.*

Proof. Induct on the query sequence. Identical previous queries give identical pools because representatives are fixed. Let B_* be the unresolved unqueried leaf of smallest priority. Every ancestor has bound at least $u_{B_*}(T) \geq c$ by Equation (25), so none can be pruned. Until B_* is reached, the frontier contains a node on its path with key at most E_{B_*}. Hence a larger-priority leaf cannot be queried first. Smaller-priority covered leaves are discarded by their exact leaf bounds; processing them or loose internal nodes adds no threshold. The next query is therefore B_* in both procedures, completing the induction. Once no unresolved unqueried leaf remains, the finite tree is pruned or subdivided into discarded leaves and terminates without another query. □

Uniformity under adaptive threshold sharing

Lemma 12 (Uniformity given the previous branch queries)**.** *Let $\mathcal{H}_q$ contain the identities and canonical responses of the first q branch queries, together with the resulting threshold pool. Condition on the fixed endpoint data and initialization, but do not include unreached priority values or internal heap events in $\mathcal{H}_q$. At every active branch-query step, the next queried leaf is uniform over the unresolved unqueried family.*

Proof. The independent continuous identically distributed priorities induce a uniform permutation, also uniform when restricted to the fixed initial candidate set $\mathcal{C}$. Suppose its unresolved restriction is uniform conditional on $\mathcal{H}_q$. Its first identifier is uniform; conditional on that identifier, the remaining order stays uniform. Fixed endpoint responses determine all subsequent deletions from $\mathcal{H}_q$ and the selected identifier, independently of that remaining order. Removing the selected leaf and restricting to the undeleted subset therefore preserves uniformity. Induction and Lemma 11 prove the claim. The information conditioned on here is essential: internal heap events would add information about the remaining priority order. □

Theorem 13 (Cover-sensitive implicit branch discovery)**.** *Assume $\max_{B \in \mathcal{B}} f_B < c$ and an unlimited search budget. For $B \in \mathcal{C}$, let $I_B = \{t : F_B(t) < c\}$ and let P be the minimum piercing number of these intervals. If $K \geq 1$, the implicit search satisfies*

$$Q_b \leq K, \qquad \mathbb{E}Q_b \leq \min\{K,\ 2P[1 + \ln(K/P)]\}.$$

It adds at most Q_b thresholds. The same high-probability branch-query bound as for uniform flat discovery applies. If $K = 0$, it makes no branch query.

Proof. Strict descent places each fixed canonical representative in its interval I_B. By Lemmas 11 and 12, queried identities have the uniform flat procedure's law, and each added threshold removes exactly the intervals containing it. These are all the hypotheses of Theorem 3, proved in Section A.5. Its harmonic expectation bound and Equation (15) therefore apply to the same K candidates and piercing number P. Each query removes itself and adds one threshold, giving the deterministic bounds and the $K = 0$ case. □

Tree work and feasibility counts A tree node needs expansion only if it contains a branch unresolved by t_0. Counting the ancestors of those K_0 branches therefore bounds search work.

Proposition 14 (Search work along initially unresolved paths)**.** *Let E be the number of expanded internal nodes and V the number of generated nodes, including the root. For the binary tree,*

$$E \leq hK_0, \qquad V \leq 1 + 2hK_0.$$

In particular the looser convention $E \le 1 + hK_0$ is also valid. The priority queue has at most $1 + hK_0$ entries. At most one exponential increment and one winning-child choice are generated per binary expansion, in addition to the root draw.

Proof. An expanded node has $U_N(T) \ge c$, so retention of t_0 and Equation (25) imply that it contains a leaf in $\mathcal{C}_0$. Each such leaf has at most h internal ancestors; their union contains all expanded nodes, proving $E \le hK_0$. Each expansion creates at most two children and increases the queue size by at most one. Hence $V \le 1 + 2E$ and the queue size is at most $1 + E$. The split rule gives the stated random-draw count. □

The possible gap between the two sides of the min–max inequality in Equation (25) is why tree work uses K_0. For cardinality retention, r undecided groups and q remaining selections give $n_N = \binom{r}{q}$; partition quotas give $n_N = \prod_j \binom{r_j}{q_j}$, with invalid quotas assigned zero. These counts depend only on feasibility.

Constructing a node bound on generation and reevaluating it once on popping uses at most $2V \le 2 + 4hK_0$ bound evaluations. Processing all $|T_0| + Q_b$ shared thresholds each time costs at most

$$2(1 + 2hK_0)(|T_0| + Q_b)$$

threshold-completion evaluations; caching can reduce repetitions. Queue work is $O(V \log(1+V))$, charged separately from exact leaf solves.

Consistent numerical upper bounds For fixed numerical upper bounds $\overline{F}_B(t)$ that do not depend on query history, use consistent leaf and node bounds

$$\overline{u}_B(T) = \min_{t \in T} \overline{F}_B(t), \qquad \overline{U}_N(T) = \min_{t \in T} \max_{B \in \mathcal{B}_N} \overline{F}_B(t).$$

More generally, node bounds may dominate every descendant leaf upper bound while remaining below the retained single-threshold completion bound. If $F_B \le \overline{F}_B \le F_B + \epsilon$ and each fixed representative lies in $\{F_B < c - \epsilon\}$, Corollary 7 applies to these effective intervals. Define candidate and critical sets using the envelopes and the cover at the shifted level. Consistent screening preserves the query law; enclosing bounds make the final comparison sound.

A.9 The computation required to identify unresolved branches

A small sufficient threshold set need not make it easy to identify which branches a given set still fails to resolve. We study that selection problem under cardinality constraints, separately from optimizing the endpoint risk. Complementary exact-sum reductions establish hardness for two-scenario selection (Kasperski & Zieliński, 2017); our linked CVaR hinges also preserve a common threshold that establishes descent with constant margin. For a supplied endpoint threshold set T and comparison level c, define the branches left unresolved by every threshold:

$$\mathcal{C}(T) = \{B : F_B(t) \ge c \text{ for every } t \in T\}.$$

The operation $\mathrm{NextUnresolved}(T)$ returns a member or EMPTY; pool coverage is the decision $\mathcal{C}(T) = \varnothing$. Inputs are rational and binary encoded, with retained cardinality variable.

The complexity statement concerns rational inputs encoded in binary. A problem is in NP when a proposed yes-answer has a certificate checkable in polynomial time; coNP contains the complements of such problems. The result below shows that checking whether a supplied pair of thresholds resolves every branch is coNP-complete, even though one newly queried threshold can establish descent.

Theorem 15 (Two-threshold pool verification)**.** *For rational group-deletion CVaR with two observations per group, tail mass $3/4$, and free cardinality retention, deciding*

$$\max_B \min_{t \in \{0,1\}} F_B(t) < c$$

is coNP-complete. The hardness reduction produces instances with

$$V_{\mathrm{D}} \le V_{\mathrm{H}} \le c - \tfrac{1}{10}, \qquad P(\mathcal{B}, c) = 1.$$

In every constructed instance, a minimizing threshold of any feasible branch bounds every branch below the current value by at least $1/10$. The construction admits globally positive, convex, smooth observation losses with curvature at most $24/5$, for which the endpoint is a relaxed-gradient update from a current value c.

A CVaR realization of complementary subset scores The construction makes thresholds 0 and 1 test opposite sides of a target subset mean. Let $w_i \in [0, 1/10]$ be the item values, $\bar{w} \in [0, 1/10]$ the target mean, and k the number to retain. The following low and high losses and comparison level c implement those two tests:

$$\ell_i^{\text{low}} = \tfrac{1}{2} - 2\bar{w} + 2w_i, \qquad \ell_i^{\text{high}} = 2 - w_i, \qquad c = \tfrac{5}{3} - \tfrac{2}{3}\bar{w}. \tag{26}$$

The low and high losses lie in $[3/10, 7/10]$ and $[19/10, 2]$, respectively. With zero regularizer, k groups give tail count $3k/2$. Writing $\bar{w}_B = k^{-1}\sum_{i\in B} w_i$, substitution at the two thresholds gives $F_B(0) = \frac{2}{3}(\frac{5}{2} - 2\bar{w} + \bar{w}_B)$ and $F_B(1) = 1 + \frac{2}{3}(1 - \bar{w}_B)$. Hence

$$F_B(0) - c = \tfrac{2}{3}(\bar{w}_B - \bar{w}), \qquad F_B(1) - c = \tfrac{2}{3}(\bar{w} - \bar{w}_B). \tag{27}$$

Both responses are at least c exactly when $\bar{w}_B = \bar{w}$. Equivalently,

$$\max_B \min\{F_B(0), F_B(1)\} = c - \tfrac{2}{3}\min_B |\bar{w}_B - \bar{w}|. \tag{28}$$

Both scores are evaluations of the same CVaR function.

Complexity reduction and membership Given positive integers $a_1, \ldots, a_n$ and target $b \in [0, A]$, $A = \sum_i a_i > 0$, in 0–1 exact sum (Karp, 1972), append n zeros and require $k = n$ out of $G = 2n$ entries. Padding any original subset with zeros, or discarding selected zeros, proves equivalence with exact-cardinality sum b. Set

$$w_i = \frac{a_i}{10A} \quad (1 \le i \le n), \qquad w_i = 0 \quad (n < i \le 2n), \qquad \bar{w} = \frac{b}{10An}.$$

These quantities lie in $[0, 1/10]$ and have polynomial binary encoding length, as do Equation (26)'s losses and level. By Equation (27), an uncovered set exists exactly when the exact-sum instance is feasible, proving NP-hardness of noncoverage. For arbitrary rational losses in the class, a set B witnesses noncoverage: its cardinality and two rational responses are verifiable in polynomial bit complexity. Thus coverage is coNP-complete. The reduction uses variable k and binary exact sum; constant k permits polynomial enumeration. The pool gap in Equation (28) can be small, while the true decrease margin below is constant.

A common threshold and a feasible probe For $t \in [3/10, 7/10]$ and $x = t - 1/2$, all high losses exceed t. Using $[z]_+ = (z + |z|)/2$ in Equation (26) cancels the linear terms in x and $w_i - \bar{w}$, giving

$$F_B(t) - c = -\tfrac{1}{6} + \frac{1}{3k}\sum_{i\in B} |2(w_i - \bar{w}) - x|\,. \tag{29}$$

Both terms inside the difference lie in $[-1/5, 1/5]$, so $\max_B F_B(t) \le c - 1/30$ throughout this interval. At its midpoint,

$$F_B(1/2) - c = -\tfrac{1}{6} + \frac{2}{3k}\sum_{i\in B} |w_i - \bar{w}| \le -\tfrac{1}{10}$$

so $V_{\text{D}} \le V_{\text{H}} \le c - 1/10$ and the common midpoint gives piercing number one. Every branch minimizer lies between its extreme low losses: the slope is $-1/3$ below that range and $1/3$ just above it, and convexity excludes more distant minima. All low losses lie in $J = [1/2 - 2\bar{w},\ 7/10 - 2\bar{w}]$, of width $1/5$. For any minimizing threshold $t \in J$, every absolute difference in Equation (29) is at most $1/5$, giving common margin $1/6 - 1/15 = 1/10$.

Thus one feasible branch solve and a shared completion establish descent. The completion selects the largest k group hinge scores in polynomial time. This probe, or a midpoint sweep, supplies new information that resolves the update while the supplied-pool operation retains its hardness.

A smooth relaxed-gradient embedding To show that this endpoint can arise from an ordinary relaxed-gradient step, interpolate from equal starting losses to the constructed endpoint. Let ℓ_j^* denote any endpoint loss from Equation (26). For a scalar parameter $\theta \in \mathbb{R}$, define

$$\ell_j(\theta) = c - 2\theta + (2 + \ell_j^* - c)\theta^2. \tag{30}$$

At zero the losses equal c with derivative -2; at one they equal ℓ_j^*. The ranges $c \in [8/5, 5/3]$, $\ell_j^* \in [3/10, 2]$ give quadratic coefficients in $[19/30, 12/5]$, hence convexity, curvature at most $24/5$, and global minimum at least $8/5 - 30/19 = 2/95 > 0$. CVaR's translation equivariance and positive homogeneity under scale θ^2 give

$$f_B(\theta) = c - 2\theta + (2 + f_B(1) - c)\theta^2.$$

The same formula holds for V_{D} and V_{H} with their respective endpoint values, including zero by continuity. Thus all these functions are smooth, $V_{\mathrm{D}}(0) = V_{\mathrm{H}}(0) = c$, and $V'_{\mathrm{H}}(0) = -2$. A relaxed-gradient step of size $1/2$ sends zero to one with true decrease at least $1/10$, preserving the supplied-pool reduction.

A preliminary probe and subsequent uniform discovery After one deterministic feasible probe and its sweep, condition on the remaining count K' and cover P'. Fresh independent priorities and Theorem 3 give

$$\mathbb{E}[Q_b \mid \text{probe information}] \le 1 + \min\{K',\, 2P'[1 + \log(K'/P')]\} \quad (K' > 0).$$

For $K' = 0$, the count is one. Charge the probe sweep and one sweep per subsequent query; a finite node budget stops a prefix before separately charged fallback.

B Variational structure and exact risk evaluation

The threshold method relies on the distinction between evaluating a retained set and optimizing over all retained sets. We derive both operations from CVaR duality, then show how cardinality and matching constraints lead to different exact algorithms.

B.1 Further connections to related work

Differentiation and relaxed learning. Optimization layers enable gradients through continuous solutions (Amos & Kolter, 2017; Agrawal et al., 2019; Blondel et al., 2022). Blackbox and perturbation methods extend learning signals to combinatorial mappings (Vlastelica et al., 2020; Berthet et al., 2020), allowing prediction models to be trained for downstream decisions (Wilder et al., 2019). Nonsmooth implicit differentiation addresses the calculus and convergence issues that arise beyond smooth solution maps (Bolte et al., 2021; Grazzi et al., 2024). These advances support proposing parameter updates; checking the finite change of the original optimized risk additionally requires endpoint information, including at nondifferentiable points.

Risk and adaptive information. CVaR's variational and dual forms (Rockafellar & Uryasev, 2000; 2002) lead to a common-threshold representation under mixture uncertainty (Zhu & Fukushima, 2009). Risk-sensitive learning develops optimization and statistical guarantees for tail risks and group robustness (Soma & Yoshida, 2020; Levy et al., 2020; Sagawa et al., 2020). DORO modifies distributionally robust risk to tolerate outliers, including a CVaR instance (Zhai et al., 2021). Interval-CVaR trims both loss tails and leads to stochastic difference-of-convex value-function optimization (Liu & Pang, 2022). Near-optimal stochastic algorithms cover group DRO, subpopulation fairness, and empirical CVaR, with a group-DRO oracle lower bound (Soma et al., 2025). These guarantees concern learning an objective; our query bounds concern deciding the sign of one prescribed finite update over a structured family of retained sets.

Federated group DRO uses a scalar threshold to express the average top-K client losses and updates that threshold alongside model parameters to reduce communication (Guo & Yang, 2024). Here a threshold optimized for one retained set supplies simultaneous upper bounds on other retained sets at a fixed endpoint. The resulting objects are sublevel intervals and a cover-discovery process. Branch and bound (Land & Doig, 1960) and safe screening (Fercoq et al., 2015; Ndiaye et al., 2017) provide the enclosing-bound principle; sharing adds a response that can remove many candidates at once. Smoothing (Nesterov, 2005) addresses the complementary task of optimizing a nonsmooth objective.

This sharing raises an acquisition question: a small threshold cover need not be cheap to find. Uncertain-value selection (Feder et al., 2003; Gupta et al., 2011) and basis verification (Diwan et al.,

2026) distinguish the information sufficient for a decision from the queries needed to obtain it. Covering and robust selection make information discovery part of the optimization task (Megow & Schlöter, 2023; Chen et al., 2026). Online interval hitting prices coverage on a known point universe (Even & Smorodinsky, 2014); oracle-based covering constructs ϵ-nets using uncovered sets (Mustafa, 2019). Our interface instead queries an unresolved branch for a fixed minimizing representative and shares its responses. Section A.5 bounds this discovery cost by the offline threshold cover.

Finally, finite precision can keep a comparison unresolved even after useful branches are found. Inexact-oracle analyses relate accuracy to optimization progress (Devolder et al., 2014; Cartis et al., 2022); verification, training with guaranteed bounds, and learning-augmented pruning use enclosing bounds for their respective guarantees (Bunel et al., 2020; Sosnin et al., 2026; Li & Hai, 2026). Our analysis ties required resolution to the finite-step sign and charges both shared queries and the computation that executes them.

B.2 Preliminary value-function facts

For every B, t, $\min_u F_B(\theta, u) \le F_B(\theta, t) \le \max_C F_C(\theta, t)$. Maximizing the first expression and minimizing the last proves $V_{\mathrm{D}} \le V_{\mathrm{H}}$. If the differentiable gap $\Gamma = V_{\mathrm{H}} - V_{\mathrm{D}}$ vanishes at an interior point θ_0, its nonnegativity gives a local minimum there. Thus $\nabla\Gamma(\theta_0) = 0$ and $\nabla V_{\mathrm{H}}(\theta_0) = \nabla V_{\mathrm{D}}(\theta_0)$ whenever both gradients exist.

Subtracting $V_{\mathrm{D}} = V_{\mathrm{H}} - \Gamma$ at the two endpoints gives Equation (6). A valid endpoint bound $\overline{H}_y < L_0$ proves descent. With exact values, $V_{\mathrm{H}}(y) < V_{\mathrm{D}}(\theta)$ is equivalent to $V_{\mathrm{H}}(\theta) - V_{\mathrm{H}}(y) > \Gamma(\theta)$, the common-threshold case of Sections A.3 and 3.

B.3 A strongly convex inner example

The ambiguity of a relaxed decrease also occurs when threshold minimization is strongly convex. We construct two quadratic branch functions with the same relaxed risk q but opposite exact risks h_σ. For initial gap $\delta > 0$ and scalar model parameter x, let $q(x) = \delta - x$, $h_\sigma(x) = \sigma x$, $\sigma \in \{-1, 1\}$, and

$$a_\sigma(x) = \sqrt{q(x) - h_\sigma(x)}, \qquad F^\sigma_\pm(x, t) = h_\sigma(x) + (t \pm a_\sigma(x))^2.$$

On $(-\delta/8, 3\delta/8)$ the square root is positive, each function is smooth with threshold curvature two, and its branch minimum is h_σ. Since $\max\{(t-a)^2, (t+a)^2\} = t^2 + a^2 + 2a|t|$ uniquely minimizes at zero, $V^\sigma_{\mathrm{D}} = h_\sigma$ and $V^\sigma_{\mathrm{H}} = q$. At zero both instances have relaxed value and gap δ, derivative -1, threshold zero, and equal active minimax weights. The step to $x = \delta/4$ gives opposite exact changes, realizing the obstruction with strongly convex inner problems.

B.4 CVaR duality, the MILP, and the exact hull relaxation

Finite-distribution CVaR To obtain an exact discrete solver, first rewrite CVaR as a maximum over observation weights. For probabilities $p_i \ge 0$ summing to one and losses ℓ_i, introduce $u_i \ge 0$ for the loss exceeding threshold t and multiplier $q_i \ge 0$ for each constraint $u_i \ge \ell_i - t$. The resulting linear program (LP) and its Lagrangian $\mathcal{L}$ are

$$\min_{t\in\mathbb{R},\, u\ge 0} \; t + \frac{1}{\beta}\sum_i p_i u_i \quad \text{subject to } u_i \ge \ell_i - t,$$
$$\mathcal{L} = t\Big(1 - \sum_i q_i\Big) + \sum_i q_i \ell_i + \sum_i u_i\Big(\frac{p_i}{\beta} - q_i\Big).$$

Minimizing over unrestricted t and nonnegative u yields

$$\max_q \sum_i q_i \ell_i \quad \text{subject to} \quad q_i \ge 0, \quad \sum_i q_i = 1, \quad q_i \le p_i/\beta. \tag{31}$$

The primal is feasible and finite, and $q = p$ is dual feasible for $\beta \le 1$. Strong duality gives the risk-envelope representation, meaning that CVaR is the largest weighted loss over the displayed feasible weights (Rockafellar & Uryasev, 2000; 2002).

For m equally weighted retained observations, write $c = \beta m = j + \rho$, $j = \lfloor c \rfloor$ and $0 \le \rho < 1$ its integer and fractional parts, and sort $\ell_{(1)} \ge \cdots \ge \ell_{(m)}$. Filling capacities in this order gives

$$f_B - R = \frac{\sum_{i=1}^{j} \ell_{(i)} + \rho \ell_{(j+1)}}{c}, \qquad t_B = \ell_{(\lceil c \rceil)}.$$

The fractional term is omitted when $\rho = 0$; at integral $c < m$, the interval to the next order statistic is also optimal. This supplies both the exhaustive reference and each queried branch.

Exact structured deletion as a MILP The binary indicator z_g records whether group g is retained. Let $Z \subset \{0,1\}^G$ encode the permitted choices of exactly k groups, $P = \mathrm{conv}(Z)$ their convex hull, also called the incidence hull because each vector encodes group membership, and $g(i)$ identify observation i's group. Equal group size s_g gives $m = ks_g$ and $p_i = z_{g(i)}/m$. Maximizing Equation (31) over these retention vectors gives

$$V_{\mathrm{D}}(\theta) - R(\theta) = \max_{z \in P \cap \{0,1\}^G,\ q \ge 0} \Big\{ \sum_i q_i \ell_i(\theta) : \sum_i q_i = 1, \quad q_i \le z_{g(i)}/(\beta m) \Big\}. \tag{32}$$

At fixed θ, the losses and denominator are coefficients, q are dual weights, and only the group indicators are discrete.

Why the continuous relaxation equals V_{H} The affine map $z \mapsto p(z)$ sends P to the convex hull of distributions over retained observations. Relaxing $z \in Z$ to $z \in P$ therefore gives

$$\max_{p \in \mathrm{conv}\{p(z) : z \in Z\}} \min_t \left[t + \frac{1}{\beta} \sum_i p_i (\ell_i - t)_+ \right].$$

The probability domain is compact and convex; the objective is affine in p and continuous convex in t. A minimizer lies in $[\min_i \ell_i, \max_i \ell_i]$, since the exterior slopes are $1 - 1/\beta \le 0$ and one. Sion's theorem (Sion, 1958) interchanges the extrema on these compact domains. At fixed t, the affine maximum over the incidence hull is attained by a feasible binary vector. The remaining minimization is the common-threshold risk, so the relaxed LP equals $V_{\mathrm{H}} - R$.

For cardinality retention, $\{z \in [0,1]^G : \sum_g z_g = k\}$ is integral: two fractional coordinates permit opposite perturbations, while a single fractional coordinate contradicts the integral sum. Balanced retention is a product of these polytopes, one per partition, so its LP is also the exact incidence hull. The number of retained groups matters: at tail mass $3/4$, groups $\{0,3\}$ and $\{2,2\}$ separately have risk two but their uniform union has risk $7/3$. A union of permitted cardinalities defines a correspondingly different risk model.

Relaxed proposal directions The relaxed dual domain is independent of θ. With differentiable observation losses and regularizer, an optimal vector q^* supplies $g_H = \sum_i q_i^* \nabla \ell_i(\theta) + \nabla R(\theta)$, the value gradient at differentiable points and a value-function subgradient at ties (Danskin, 1967). In either case the checker evaluates the resulting finite displacement.

B.5 Weighted CVaR families that attain harmonic discovery

The logarithm in the uniform-query bound reflects repeated partial removal of nested intervals. We construct P disjoint groups of n nested intervals each. A query removes a uniformly distributed fraction of its group, so the expected number of queries per group is the harmonic sum $H_n = \sum_{j=1}^{n} 1/j$.

Proposition 16 (A family attaining the uniform-query bound)**.** *For every pair of positive integers P, n, there are $K = Pn$ weighted two-point CVaR branches, at tail mass $1/2$, whose candidate threshold-cover number is P and whose uniformly selected canonical queries use exactly PH_n branch solves in expectation. The branches have positive probabilities and nonnegative losses.*

Proof. Let $C = P(n+3)$ be the comparison level. Place cluster r at right endpoint $p_r = C - (r-1)(n+3)$, so consecutive clusters are separated. For $r = 1, \ldots, P$ and $i = 2, \ldots, n+1$, choose low and high losses $a_{r,i}, b_{r,i}$ and their probabilities as

$$a_{r,i} = p_r - i, \qquad b_{r,i} = \frac{iC - p_r}{i-1}, \qquad \Pr(a_{r,i}) = \frac{i+1}{2i}, \quad \Pr(b_{r,i}) = \frac{i-1}{2i}.$$

The probabilities are positive and sum to one. Since $p_r \geq n+3$, we have $a_{r,i} \geq 2$, $b_{r,i} \geq C$, and $b_{r,i} - a_{r,i} = i[1 + (C - p_r)/(i-1)] > 0$. The response $F_{r,i}(t) = t + 2\mathbb{E}[(\ell - t)_+]$ has successive slopes $-1, 1/i, 1$, hence unique minimizer $t_{r,i} = p_r - i$. Its value and strict sublevel are

$$f_{r,i} = \frac{a_{r,i}}{i} + \frac{i-1}{i} b_{r,i} = C - 1, \qquad I_{r,i}(C) = (p_r - i - 1, p_r).$$

The roots follow from the two adjacent slopes. Within each cluster the intervals are nested. Their unions lie in $(p_r - n - 2, p_r)$ and are separated by the center spacing $n+3$, so the piercing number is P.

Start with valid bounds $u^0_{r,i} = C$. The representative $p_r - i$ lies in $I_{r,j}$ exactly when $j \geq i$, and in no other cluster. A cluster with m remaining consecutive smallest indices therefore has a uniform query rank, leaving a uniform count in $\{0, \ldots, m-1\}$. Let T_m be the expected number of further selections from a cluster containing m unresolved intervals. One query leaves each size from zero to $m-1$ with equal probability, giving

$$T_0 = 0, \qquad T_m = 1 + \frac{1}{m} \sum_{q=0}^{m-1} T_q, \qquad T_m - T_{m-1} = \frac{1}{m}.$$

Thus $T_m = H_m$. Interleaving preserves each cluster's conditional rank law, and linearity gives $\mathbb{E}Q_b = \sum_{r=1}^{P} T_n = PH_n$. □

This attains $P \log(eK/P)$ for the specified policy; Theorem 4 gives the separate all-policy obstruction at $P_{\mathcal{C}} = K$.

Order dependence within one cluster For $i = 1, \ldots, M$, take losses $C - i, C$ with probabilities $(i+1)/(2i), (i-1)/(2i)$; $i = 1$ has one effective observation. Then

$$F_i(t) = \begin{cases} 2C - i - 1 - t, & t \leq C - i, \\ C + (t - C)/i, & C - i \leq t \leq C, \\ t, & t \geq C. \end{cases}$$

Its minimum is $C - 1$ at $t_i = C - i$, with interval $(C - i - 1, C)$. The initial sweep $t_0 = C$ gives equal bounds C. Resolving largest-bound ties in order $M, \ldots, 1$ requires M solves because $F_j(t_i) = C + i - j - 1 \geq C$ for every remaining $j < i$. Uniform selection uses H_M solves in expectation, while choosing the smallest index first gives one query.

B.6 Using threshold derivatives to guide the search

Threshold derivatives identify which direction can lower a convex response, allowing interval bisection without an exact branch minimization. Let $c = L_0$. We call an evaluation a signed sweep when it returns $(F_i(t), g_i(t))$ for every branch, where $g_i(t) \in \partial F_i(t)$ is a fixed one-sided subgradient with respect to t. A subgradient defines a line through $(t, F_i(t))$ lying below the convex response, so its sign identifies a direction containing a minimizer. Assume each minimum has a representative in $[a, b]$, of width $R = b - a$, and every F_i is S-Lipschitz there, meaning $|F_i(t) - F_i(u)| \leq S|t - u|$ for $t, u \in [a, b]$. Here the scalar R measures bracket width rather than regularization. Sweep the midpoint of each active bracket $[\ell, r]$, removing branches with $F_i(t) < c$. Positive subgradients route left, negative ones right; convexity preserves a minimizer in the chosen half. A zero subgradient gives $f_i = F_i(t)$, establishing nondecrease if $f_i \geq U_0$ and retaining baseline uncertainty otherwise. The supporting-line lower bound is

$$f_i \geq F_i(t) + \min\{g_i(t)(\ell - t),\ g_i(t)(r - t)\}.$$

The breadth-first implementation updates global upper bounds after each sweep, retains bracket-specific lower bounds, and returns *unknown* when its budget ends.

Proposition 17 (A deterministic bound from signed sweeps)**.** *Let $\mu > 0$ be the available risk separation and P the minimum number of points intersecting all branch sublevel intervals. If $\max_i f_i \leq c - \mu$, midpoint signed search establishes descent without branch solves in at most*

$$Q_s \leq \sum_{d=0}^{D} \min\{2^d, P\} \leq 4P + P \log_2^+ \frac{RS}{P\mu}, \qquad D = \lceil \log_2 \max\{1, RS/\mu\} \rceil$$

sweeps, where P is the piercing number of the nonempty branch sublevels. Here $\log_2^+(x) = \log_2 \max\{1, x\}$ *for* $x \geq 0$.

Proof. Before removal, a branch's sublevel within $[a, b]$ lies in its routed cell: a midpoint with value at least c separates that sublevel to the side indicated by its subgradient. Distinct active cells at depth d thus contain disjoint unresolved sublevels.

Clipping preserves their piercing number. Any subfamily with a common point still intersects jointly after adding $[a, b]$: each interval meets the minimizer bracket, so the interval Helly property applies. Original disjoint intervals remain disjoint. The clipped packing number is consequently P, giving at most $\min\{2^d, P\}$ active cells.

A depth-d midpoint is within $R2^{-d-1}$ of its branch minimizer, so $F_i(t) \leq f_i + SR2^{-d-1}$. At depth D this is at most $c - \mu/2 < c$, removing all remaining branches. Summing cell counts proves the first bound. Through depth $\lfloor \log_2 P \rfloor$ the geometric sum is less than $2P$; each remaining depth contributes at most P. Using $D \leq \log_2 \max\{1, RS/\mu\}+1$ gives the second bound, with $RS/\mu < P$ bounded directly by $4P$. □

For empirical CVaR, the right derivative is $1 - \#\{i \in B : \ell_i > t\}/(\beta m)$, and the observed-loss range contains a minimizing quantile. On the nested one-cluster example only the right child stays active, giving $\max\{1, \lfloor \log_2 M \rfloor\}$ sweeps. The smooth private-region lower bound also holds for signed sweeps: outside a branch's closed private interval, both its response and its fixed one-sided slope are independent of its bit, so a query exposes at most one coordinate.

B.7 Tree completion bounds and total search work

A node bound should cover all its possible retained sets without evaluating them separately. At a fixed threshold, each group contributes its sum of excess losses, so cache $a_g(t) = \sum_{i \in g} (\ell_i(y) - t)_+$. A cardinality node fixes included groups I and undecided groups R, from which it must retain $k - |I|$. Write $\mathrm{Top}_{k-|I|}(R, t)$ for the $k - |I|$ undecided groups with largest scores at t. Selecting them maximizes the response over all branches below the node:

$$U_N(t) = t + R(y) + \frac{1}{\beta m}\left[\sum_{g \in I} a_g(t) + \sum_{g \in \mathrm{Top}_{k-|I|}(R,t)} a_g(t)\right] = \max_{B \in N} F_B(y, t). \tag{33}$$

Infeasible nodes are discarded; disjoint quotas apply the same selection within each partition. The pool bound $\min_{t \in T} U_N(t)$ encloses every descendant minimum. The largest-bound queue rechecks this bound on popping a node; nodes below the comparison level remove part of the family, while a leaf supplies a minimizing threshold or a lower witness. Closed nodes, queried leaves, and the remaining frontier retain an endpoint enclosure when a budget stops the search.

Proposition 18 (Output-sensitive completion bound)**.** *Retain t_0 and write $K_0 = |\{B : F_B(y, t_0) \geq L_0\}|$. A depth-$G$ binary group-selection tree expanding only bounds at least L_0 uses at most GK_0 expansions, K_0 leaf queries, and $1 + 2GK_0$ generated nodes.*

Proof. Every expanded node contains a t_0-critical leaf by Equation (33). Apply Proposition 14 with $h \leq G$; only critical leaves can be queried. This counts nodes separately from scores, selections, and queue work. □

B.8 An exact dynamic program for disjoint equal-size groups

The exact solver distributes the CVaR tail weight among retained groups. It records how many groups and tail observations have already been selected, avoiding enumeration of all retained sets. At a fixed endpoint retain exactly k of G disjoint groups, each containing $s \geq 1$ observations, with $1 \leq k \leq G$. The $m = ks$ retained observations have probability $1/m$. For tail mass $0 < \beta \leq 1$, the tail contains $\alpha = \beta m$ observation-equivalents: j receive full weight and, if needed, one receives fractional weight ϕ. Let e_* indicate whether that fractional observation is needed:

$$\alpha = \beta m = j + \phi, \qquad j = \lfloor \alpha \rfloor, \quad 0 \leq \phi < 1, \qquad e_* = \mathbf{1}_{\{\phi > 0\}}.$$

Let R_y be the group-independent regularizer. Sort each group's losses $L_{g,1} \geq \cdots \geq L_{g,s}$ and set $S_g(r) = \sum_{a=1}^{r} L_{g,a}$, with $S_g(0) = 0$.

Lemma 19 (Tail-allocation vertices)**.** *For a retained group set B, let $\mathcal{I}(B)$ index its observations. The variable u_i below is the scaled tail weight assigned to observation i. Its CVaR numerator is the maximum of $\sum_{i\in\mathcal{I}(B)} u_i \ell_i$ over*

$$0 \leq u_i \leq 1, \qquad \sum_{i\in\mathcal{I}(B)} u_i = \alpha.$$

Every vertex has j coordinates equal to one and, when $\phi > 0$, one further coordinate equal to ϕ; all other coordinates are zero. For any fixed numbers of full and fractional coordinates assigned to a group, an optimum assigns the full coordinates to its largest losses and its fractional coordinate, if present, to the next largest loss.

Proof. Scale the CVaR dual weights by α. Two fractional coordinates permit small opposite transfers, so a vertex has at most one. The sum constraint fixes its value and the number of unit coordinates. Conversely, fixing all but one coordinate at box bounds uniquely determines the last, proving these feasible points are vertices. A linear objective on this compact polytope attains an optimum at a vertex. Permuting weights within a group preserves feasibility; swapping a larger weight onto a larger loss cannot decrease the objective, proving the stated order. □

States and transitions For one retained group, let r count its unit-weight observations and $d \in \{0, 1\}$ indicate a fractionally weighted observation. The allowed actions form Ω_ϕ, and $W_g(r, d)$ is their contribution to the tail numerator:

$$\Omega_\phi = \{(r, d) : 0 \leq r \leq s,\ 0 \leq d \leq e_*,\ r, d \in \mathbb{Z},\ r + d \leq s\},$$
$$W_g(r, d) = \begin{cases} S_g(r), & d = 0, \\ S_g(r) + \phi L_{g,r+1}, & d = 1. \end{cases}$$

The action $(0, 0)$ is retained, and $(s, 0)$ never accesses an $(s + 1)$st loss. Let $A_g(q, h, e)$ be the maximum reward after processing g groups, retaining $q \leq k$, using $h \leq j$ unit-weight observations and $e \leq e_*$ fractionally weighted observations. A state summarizes these three counts, and its reward is the unnormalized weighted loss. Initialize $A_0(0, 0, 0) = 0$; all other initial and out-of-range states are $-\infty$. Skipping or retaining group g gives

$$A_g(q, h, e) = \max \left\{ A_{g-1}(q, h, e),\ \max_{\substack{(r,d)\in\Omega_\phi \\ r\leq h,\ d\leq e}} [A_{g-1}(q - 1, h - r, e - d) + W_g(r, d)] \right\}. \tag{34}$$

After all groups, state (k, j, e_*) enforces the required retained count and tail mass. Dividing its numerator by α and adding the regularizer gives

$$V_D(y) = R_y + \frac{A_G(k, j, e_*)}{\alpha}. \tag{35}$$

Theorem 20 (Exact polynomial endpoint evaluation)**.** *The recursion Equation (34) gives Equation (35) exactly and can reconstruct an optimal retained group set and feasible CVaR dual weights. Its arithmetic-operation complexity is*

$$O(Gs \log(s + 1) + G(k + 1)(j + 1)(s + 1)).$$

Keeping only the preceding and current layers of values uses $O((k + 1)(j + 1))$ memory. Storing the maximizing previous state and action permits reconstruction with $O(G(k + 1)(j + 1))$ memory. For rational inputs, exact evaluation also has polynomial bit complexity.

Proof. By Lemma 19, an optimal allocation for any retained set is represented by the group actions. Conversely, any action sequence selecting k groups with totals (j, e_*) defines a feasible scaled CVaR vertex. Induction on g proves the recurrence: every allocation either skips g or retains it with one action, giving its two alternatives. The terminal state maximizes the numerator over all retained sets.

Backtracking maximizing predecessors from (G, k, j, e_*) recovers the groups and actions (r_g, d_g). Assign weight one to each group's first r_g losses and weight ϕ to its next loss if $d_g = 1$, then divide

by α. These nonnegative weights sum to one, obey the $1/\alpha$ cap, and attain the terminal objective. The reconstructed retained set is therefore optimal; its sorted-tail formula also supplies a canonical minimizing threshold.

There are at most $2(k+1)(j+1)$ states per layer and $2(s+1)$ retained actions per state. Sorting and prefix sums give the stated time; two value layers suffice, and stored predecessors give the reconstruction memory. Since $j \le ks \le Gs$, the state count is polynomial in the observation count. Each reward sums at most Gs rational losses and one ϕ-weighted loss, so exact arithmetic has polynomial bit length and operation count. □

Disjoint partition quotas To impose one retention quota per partition, solve each partition locally and then allocate the tail counts across partitions. For disjoint partitions $\mathcal{G}_1, \dots, \mathcal{G}_R$ with quotas k_r summing to k, let $B_r(h,e)$ be partition r's best numerator and $C_r(h,e)$ the best combined numerator for its first r partitions:

$$B_r(h,e) = A^{(r)}_{|\mathcal{G}_r|}(k_r, h, e),$$
$$C_r(h,e) = \max_{\substack{h_1+h_2=h \\ e_1+e_2=e}} \{C_{r-1}(h_1,e_1) + B_r(h_2,e_2)\}.$$

All partitions use the global fractional coefficient ϕ. Set $C_0(0,0) = 0$ and all other states to $-\infty$; the answer is $R_y + C_R(j, e_*)/\alpha$. Every feasible allocation uniquely splits its counts across partitions, proving the merge recurrence. Local and merge predecessors reconstruct its retained set. The arithmetic cost is

$$O\left(Gs\log(s+1) + \sum_{r=1}^{R} |\mathcal{G}_r|(k_r+1)(j+1)(s+1) + R(j+1)^2\right).$$

This is polynomial even when the partition count is input, avoiding a Cartesian product of quota counters. Balanced retention is the two-partition case.

Boundary cases and scope For $\phi = 0$, fractional actions and the terminal flag are zero. At $\beta = 1$, $j = ks$ forces full weight on every retained observation: the answer is $R_y + (ks)^{-1} \sum_{g \in B^*} S_g(s)$, where B^* maximizes group totals under the quotas; the smallest retained loss is a valid threshold. When $j = 0$, one observation receives weight ϕ and the other $k-1$ groups may receive none. The retained action $(0,0)$ ensures exactly-k semantics throughout. Ties and negative losses use the same recurrence with exact tail-weight counts and unreachable value $-\infty$.

The theorem covers disjoint equal-size groups, equal retained probabilities, and cardinality or disjoint partition quotas. Its optimized endpoint $\max_B \min_{t \in \mathbb{R}} F_B(t)$ differs from the supplied-pool operation $\max_B \min_{t \in T} F_B(t)$ in Section A.9. The latter can remain at the comparison level even when the exact endpoint has a constant descent margin; the DP can establish the decrease by evaluating the endpoint directly.

B.9 Coupled row and column retention

Row and column constraints couple the retained groups: choosing one cell excludes the other cells in its row and column. Index n^2 disjoint groups by $(a,b) \in [n]^2$, where $[n] = \{1, \dots, n\}$, each with s observations. For a permutation π assigning a distinct column to each row, the retained set $B_\pi = \{(a, \pi(a)) : a \in [n]\}$ retains one group per row and column, giving $m = ns$ observations. Evaluate all losses and the regularizer at y, and allow $0 < \beta \le 1$.

The incidence hull and relaxed proposal To propose a relaxed update, replace each binary cell choice by a nonnegative weight z_{ab} while retaining row and column sums of one. These weights form the convex hull of permutation matrices, the doubly stochastic polytope (von Neumann, 1953)

$$\mathcal{P}_n = \left\{ z \ge 0 : \sum_b z_{ab} = 1 \ (a \in [n]), \quad \sum_a z_{ab} = 1 \ (b \in [n]) \right\}.$$

To see integrality, every row or column incident to a fractional entry has at least two, since its sum is one. Their bipartite graph contains an even cycle; small alternating additions and subtractions

preserve feasibility and express the matrix as a nontrivial midpoint. Vertices are therefore integral permutation matrices. Conversely, a permutation's zero entries force the same support in any convex decomposition, making it a vertex. Compactness gives the claimed hull.

Let $v = 1, \dots, s$ index observations within a cell, q_{abv} their CVaR weights, and $\alpha = \beta ns$ the tail count. The exact problem restricts z to binary choices; the relaxed problem allows every $z \in \mathcal{P}_n$:

$$V_{\mathrm{D}}(y) - R(y) = \max_{z \in \mathcal{P}_n \cap \{0,1\}^{n \times n},\ q \geq 0} \left\{ \sum_{abv} q_{abv} \ell_{abv}(y) : \sum_{abv} q_{abv} = 1,\ q_{abv} \leq z_{ab}/\alpha \right\},$$

$$V_{\mathrm{H}}(y) - R(y) = \max_{z \in \mathcal{P}_n,\ q \geq 0} \left\{ \sum_{abv} q_{abv} \ell_{abv}(y) : \sum_{abv} q_{abv} = 1,\ q_{abv} \leq z_{ab}/\alpha \right\}.$$

Fixing a permutation gives the first equality by CVaR duality. For the second, apply Section B.4's compact-domain minimax argument to weights $z_{ab}/(ns)$; the fixed-threshold affine maximum selects a permutation, giving $\min_t \max_\pi F_{B_\pi}(y, t)$.

An integral incidence hull can have a positive risk gap. At $n = s = 2$, $\beta = 3/4$, and $R = 0$, give diagonal groups losses $\{0, 3\}$ and off-diagonal groups $\{2, 2\}$. Both retained sets have risk two; on $[0, 2]$ their responses are $2 + t/3$ and $8/3 - t/3$. Opposite slopes at their intersection $t = 1$ establish $V_{\mathrm{D}} = 2$ and $V_{\mathrm{H}} = 7/3$.

An assignment completion with exact descendant counts Fix the row order. A node stores assigned distinct columns and a nonempty allowed set S for the current row. With r rows remaining and unused columns C, require $S \subseteq C$; later rows can use every column left after that choice. It has $n_N = |S|(r-1)!$ descendants: each choice in S leaves $(r-1)!$ bijections. Split S into two fixed balanced subsets, whose counts sum to n_N, and compress a singleton choice into the prefix. This gives a fixed binary tree of $n!$ leaves, independent of priority randomness, with

$$h \leq \sum_{r=1}^{n} \lceil \log_2 r \rceil \leq \log_2(n!) + n.$$

Cache $w_{ab}(t) = \sum_{v=1}^{s} [\ell_{abv}(y) - t]_+$. The prefix A_N contains the cells already chosen at node N. To bound the remaining choices, maximize their total score over bijections σ from remaining rows to unused columns C, with the current row restricted to S. Denote this fixed-threshold maximum by $M_N(t)$:

$$M_N(t) = t + R(y) + \frac{1}{\alpha} \left[\sum_{(a,b) \in A_N} w_{ab}(t) + \max_{\substack{\sigma : \text{remaining rows} \to C \text{ bijective} \\ \sigma(\text{current row}) \in S}} \sum_{a \text{ remaining}} w_{a,\sigma(a)}(t) \right]. \quad (36)$$

The assignment is feasible since $S \neq \varnothing$ and later rows are unrestricted. A shortest-path Hungarian implementation (Kuhn, 1955; Edmonds & Karp, 1972) computes it in $O(r^3)$ arithmetic operations. For $U_N(T) = \min_{t \in T} M_N(t)$,

$$\max_{B \in \mathcal{B}_N} \min_{t \in T} F_B(y, t) \leq U_N(T) \leq M_N(t_0), \qquad t_0 \in T_0 \subseteq T,$$

with equality in the first inequality at leaves.

Corollary 21 (Matching-family discovery)**.** *Let E count expanded internal nodes and V all generated nodes, including the root. For this binary representation, the implicit-priority algorithm has the uniform unresolved-branch query law of Theorem 9. On a strict-descent instance with exact fixed representatives and unlimited budget,*

$$\mathbb{E}Q_b \leq \min\{K, 2P[1 + \log(K/P)]\} \quad (K \geq 1), \qquad E \leq hK_0, \quad V \leq 1 + 2hK_0.$$

For $K = 0$, $Q_b = 0$. Here K, P, and K_0 use the initial candidate intervals and retained threshold specified in that theorem.

Proof. The fixed tree, exact counts, leaf responses, and completion bounds satisfy Theorem 9. Its coupling gives uniform discovery, and its critical-path count gives the bounds with depth h. □

Generation-plus-pop evaluation uses at most $2 + 4hK_0$ bounds, each over at most $|T_0| + Q_b$ thresholds. Global score construction costs $O((|T_0| + Q_b)n^2 s)$, assignments cost $O((1 + hK_0)(|T_0| + Q_b)n^3)$, branch sorting costs $O(Q_b ns \log(ns+1))$, and queue operations cost $O((1+hK_0)\log(2+hK_0))$. These charge representation work for the $n!$-leaf family separately from branch queries. Cached scores reduce repeated processing; a node budget stops a prefix before separately charged exact fallback.

A native exact reference using column subsets For the exact endpoint value, the solver must choose both a matching and its tail observations. Reuse $\alpha = j + \phi$, local actions Ω_ϕ, and rewards $W_{ab}(r, d)$ from Section B.8; here r counts unit-weight observations and d indicates a fractional one. Let $A(C, q, e)$ maximize reward after assigning the first $|C|$ rows to column set C, with q full and e fractionally weighted observations. Initialize $A(\varnothing, 0, 0) = 0$ and other states to $-\infty$. For $a = |C| + 1$, $b \notin C$, and $(r, d) \in \Omega_\phi$, update

$$A(C \cup \{b\}, q + r, e + d) \leftarrow \max\{A(C \cup \{b\}, q + r, e + d),\ A(C, q, e) + W_{ab}(r, d)\}$$

when $q+r \le j$ and $e+d \le e_*$. The answer is $V_{\mathrm{D}}(y) = R(y)+A([n], j, e_*)/\alpha$. Every path chooses one unused column per row and a feasible tail allocation. Conversely, each matching's optimal tail vertex appears by Lemma 19. Induction on $|C|$ proves the value; predecessors recover the matching and observation weights.

When $ns - j < j$, count observations with weight below one instead: a local action consumes $s - r$ units, including its fractionally weighted observation when present. Every terminal allocation has $ns-j$ such observations. Replacing increment r and target j by $s-r$ and $ns-j$ therefore bijectively reindexes allocations without changing rewards or the fractional flag. With $q_* = \min\{j, ns - j\}$, this gives

$$O\big(n^2 s \log(s + 1) + n2^n(q_* + 1)(s + 1)\big)$$

arithmetic operations and $O(2^n(q_* + 1))$ states and backpointers. Zero-contribution retained actions remain available; $\phi = 0$ disables the fractional flag. At $\beta = 1$, every retained observation contributes, so assignment on group sums is a polynomial exact endpoint solver.

Connection to a classical matching objective A bipartite graph has two vertex sets, here rows and columns, with edges only between the sets. A perfect matching chooses one edge incident to each vertex. This connects matching-constrained risk to selecting the largest edge weights within such a matching.

Proposition 22 (Embedding top-weight perfect matching)**.** *For $1 \le \kappa \le n$, an exact endpoint solver for complete matching-constrained CVaR groups can recover the maximum sum of the κ largest edge weights in a perfect matching of an arbitrary balanced bipartite graph, using two observations per group and integer tail count.*

Proof. First check perfect-matching feasibility. For a feasible graph with $0 \le w_{ab} \le W$, set $C = nW + 1$. Allowed edges give losses $\{C, w_{ab}\}$ and absent edges $\{0, 0\}$. Set $\beta = (n + \kappa)/(2n)$ and $R = 0$. Any all-allowed matching has tail numerator nC plus its κ largest weights. A matching with an absent edge has at most $n - 1$ copies of C and n other positive losses bounded by W, so its entire sum, hence its tail numerator, is at most $(n - 1)C + nW < nC$. Thus every optimum uses allowed edges and recovers the source optimum as $(n + \kappa)V_{\mathrm{D}} - nC$. The construction has polynomial size and preserves polynomially bounded integer weights. □

Maximizing top-κ perfect-matching weight is polynomial-time equivalent to Exact Matching, which asks for a perfect matching of a specified total weight, for polynomially bounded positive integer weights, preserving bipartite inputs (El Maalouly, 2023, Theorem 16). Shifting nonnegative weights by one adds the same κ to every objective. The embedding varies β with κ: $\kappa = n$ is ordinary assignment, while $\kappa < n$ gives $\beta < 1$. This locates endpoint optimization within matching theory; the experiment compares its native subset solver with assignment-based descent checking.

C Numerical guarantees and experimental methods

The theoretical descent test requires valid risk bounds and comparable proposed updates. We first specify the data model and shared training rule, then derive the numerical bounds used by branch

checks, assignments, and exact dynamic programming. These derivations connect the real-valued theory to the experimental decisions; Sections D and E analyze the resulting learning paths and costs.

C.1 One controlled data-generating family

Here $N(\mu, \Sigma)$ denotes a Gaussian distribution with mean μ and covariance Σ, I_d is the d-dimensional identity matrix (with the appropriate dimension when its subscript is omitted), and $\mathrm{Unif}[a, b]$ is the uniform distribution on $[a, b]$. The synthetic data combine shifts in group covariates, group-specific regression coefficients, and two residual profiles, so different retained groups can determine the tail risk. Let d_θ be the input dimension and s the even number of observations per group. Draw $w_* \sim N(0, I_{d_\theta})/\sqrt{d_\theta}$ and normalize an independent Gaussian vector v_* to unit length. For each group, draw $c_g \sim N(0, 0.35^2 I)$, $e_g \sim N(0, 0.08^2 I)$, and $a_g \sim \mathrm{Unif}[0.8, 1.2]$. Here w_* is the shared regression coefficient, v_* a direction of group variation, c_g a covariate shift, e_g a coefficient perturbation, and a_g a residual scale. The sign σ_g is 1 on the first half of groups and -1 on the second. Groups alternate between a concentrated profile $p_g = (0.25, \ldots, 0.25, 2, \ldots, 2)$ with equal halves and a uniform profile $p_g \equiv 1.6$, starting with the concentrated profile. With independent covariate noise $\xi_{gi} \sim N(0, I)$ and observation noise $\epsilon_{gi} \sim N(0, 0.04^2)$, construct input x_{gi}, coefficient w_g, response y_{gi}, and initial parameter θ_0 as

$$x_{gi} = (\xi_{gi} + c_g)/\sqrt{d_\theta}, \qquad w_g = w_* + 0.25\sigma_g v_* + e_g,$$
$$y_{gi} = x_{gi}^\top w_g + a_g\, p_{gi}(-1)^{i-1} + \epsilon_{gi}, \qquad \theta_0 = w_* + e_0, \quad e_0 \sim N(0, 0.7^2 I).$$

Losses are $\ell_{gi}(\theta) = \frac{1}{2}(x_{gi}^\top \theta - y_{gi})^2$ and $R(\theta) = 0.015\|\theta\|^2$. The primary grid has $d_\theta = 6$, $s = 4$; each later study specifies its size. Each seed generates an independent dataset, reused across risk configurations and checking rules so that their comparisons are paired.

C.2 Matched proposal experiment

On $G = 12$ groups, delete $b \in \{4, 8, 10\}$ and retain $k = 12 - b$, either freely or with $k/2$ groups from each half. Tail masses are $\beta \in \{0.5, 0.75, 0.9\}$; the 18 configurations have 36–495 branches. To test identical proposals across checkers, first generate a common relaxed-training path. Let $g_H(\theta_j)$ be the relaxed gradient or subgradient at path index j, and let $L_{\mathrm{sample}} = \max_i \|x_i\|^2 + 0.03$ set the step scale. Starting from θ_0, use

$$\theta_{j+1} = \theta_j - \frac{0.7\, g_H(\theta_j)}{L_{\mathrm{sample}}\sqrt{1 + j/10}}.$$

At $j \in \{0, 80, 240\}$, propose $-c\, g_H/L_{\mathrm{sample}}$ for $c \in \{0.005, 0.05, 0.5, 1.5, 4, 12\}$. Twenty seeds give 6,480 endpoints. Enumeration at both endpoints supplies the reference. The checker receives the current value enclosure and relaxed threshold, using that threshold for its initial endpoint bounds. The scale L_{sample} sets the proposal size; the descent check uses feasible thresholds across CVaR tail switches.

The no-sharing control refines queried branches without sharing their minimizing thresholds. Query caps $0, 1, 2, 4$ measure coverage under limited information. The gap-only control establishes descent when the endpoint relaxation bound is below the baseline. Common baseline setup is timed separately; endpoint time includes loss evaluation, screening, scans, and queries.

C.3 Shared protocol for the larger learning studies

The larger linear-model synthetic studies use dimension 32 and share initial data and parameters across policies, with shuffled execution order per configuration. The controller below specifies their common 100-position, twelve-trial protocol, cache, numerical comparison, and complete accounting. Stopped states carry forward at zero further cost. The study-specific fallback is full MILP, decision MILP, or native DP; full-MILP calls require optimal status, an absolute difference of at most 10^{-7} between the feasible value and solver upper bound, and outward allowance 10^{-8}. Decision-MILP targets appear in Section D.9. Choices are fixed on separate development data; each study specifies its evaluation seeds. The trainable-network study in Section D.3 uses the same endpoint decision rule with its stated model, gradient, and step size.

C.4 BRANCH SELECTION AND THE COMPLETE TRAINING PROCEDURE

Probe selection and development protocol. The first branch query reuses a feasible branch that previously supplied a lower bound, avoiding a new search just to select the probe. Its stored index B_* is initialized by the first global reference at the starting parameter. After an accepted proposal, the controller stores the branch attaining the largest available lower bound at that endpoint. A global baseline refresh replaces it by that solve's maximizing branch. A rejected trial can add thresholds to the four-entry recency cache; its endpoint witness becomes the next probe only if the trial is subsequently accepted. Native solver and branch ties follow their deterministic traversal order. Selecting B_* is a stored-identifier lookup, followed by an $O(m)$ partial-selection branch evaluation on the prepared endpoint losses. The cache controls reusable thresholds, not a second selection of the probe.

The rule was developed in the synthetic pilot study using seed 0 and carried unchanged into the matching, chronological public-data, fixed-feature, and trainable-network studies. Development seeds 2099 and 4099 check the public linear and trainable-network implementations, respectively, and are excluded from their final summaries. The matching evaluation uses seeds 260–279; the network uses 4100–4109. All probe policies share the cache of thresholds from four branches; probe/root has zero expansions, and greedy and uniform have 64. The protocols fix these choices before their evaluation grids and compute heldout metrics after training.

Algorithm 2 states the flat procedure used in the query analysis: maintain one bound per branch and share each new threshold with all of them. Its displayed sweep uses exact responses. With numerical error, replace $F_C(y,t)$ by its valid upper bound, and keep the returned branch interval. The tree implementation below applies the same tests without listing the full family.

Algorithm 2 BRANCHCHECK: check a proposed endpoint y

Require: Baseline $[L_0, U_0]$, upper bounds u_B^0, query budget
1: Set $u_B \leftarrow u_B^0$, $\ell_B \leftarrow -\infty$; mark all branches unqueried
2: **while** $\max_B u_B \geq L_0$ and $\max_B \ell_B < U_0$ **do**
3: Select uniformly among unqueried branches with $u_B \geq L_0$
4: If none exists or the budget is exhausted, return *unknown*
5: Query its enclosure $[\ell, u]$ and threshold t; update bounds and mark it queried
6: Sweep t and set $u_C \leftarrow \min\{u_C, F_C(y,t)\}$ for all branches C
7: **end while**
8: Return *descent* if $\max_B u_B < L_0$; otherwise return *counterexample*

The endpoint state and its cost. The matching nodes, splits, counts, and queue rules are specified in Algorithm 4. Endpoint preparation evaluates losses and the regularizer and fixes numerical error allowance e and score-grid spacing ω under Section C.9; that proof denotes the same spacing by δ. For row a and column b, $\widehat{\ell}_i(y)$ denotes a computed loss in that cell. The hatted sum below denotes floating-point summation. Rounding the summed threshold excess upward in units of ω gives an integer score $W_{ab}(t)$, stored once per distinct threshold:

$$W_{ab}(t) = \left\lceil \widehat{\sum}_{i\in(a,b)} [\widehat{\ell}_i(y) - t]_+ / \omega \right\rceil .$$

The function $\psi_t(Z)$ in Equation (43) converts a total integer score Z into a risk upper bound at threshold t, including numerical allowances. It is nondecreasing in Z, so the same expression works for both nodes and leaves. A branch call computes minimum risk $\widehat{f}_B$ and its fixed quantile, returning the interval $[\widehat{f}_B - e, \widehat{f}_B + e]$. Each node stores an upper bound u_N and the number v_N of thresholds already included in that bound.

Updating a node after a new threshold. To update node N, $\mathsf{Bound}(N, u, v)$ takes its previous upper bound u and processed-threshold count v. Let $Z_N(t)$ be the largest feasible integer score below that node at threshold t; one update counts as one bound evaluation. For each new entry $j = v+1, \ldots, |T|$, it sums the prefix scores and solves integer assignment on the remaining rows under the allowed columns, charging one assignment per nonleaf completion. Leaves need only the

prefix sum. It returns $\min\{u, \min_{j>v} \psi_{t_j}(Z_N(t_j))\}$ with processed-threshold count $|T|$; new nodes start at $(+\infty, 0)$.

Let U_{closed} be the largest upper bound of discarded nodes and queried leaves, and L_{leaf} the largest queried-leaf lower bound. Let $\mathcal{Q}$ be the frontier, the nodes still awaiting processing. Queried leaves, closed nodes, and the frontier cover all feasible branches, so throughout the search,

$$L_{\text{leaf}} \le V_{\text{D}}(y) \le \max\{U_{\text{closed}}, \max_{N \in \mathcal{Q}} u_N\},$$

Adding thresholds preserves stored bounds; processing each threshold once avoids repeated completions. At return, merge the probe's lower bound with L_{leaf}, replacing the leaf witness only for a strictly larger lower bound. Early probe nondecrease uses upper bound $+\infty$; early root descent uses the root bound. The query trace includes rejected proposals. A range-check failure returns *unknown*.

From endpoint checks to complete training. Define $\mathsf{Classify}(L_1, U_1; L_0, U_0, \gamma)$ as descent when $U_1 < L_0 - \gamma$, nondecrease when $L_1 \ge U_0$, and unknown otherwise. For a model parameter vector x, the helper $\mathsf{Ref}(x)$ evaluates losses, runs global DP, reconstructs a maximizing retained set, and returns a lower bound, an upper bound, and the maximizing branch. Every invocation counts, including initialization, fallback, and baseline refresh; obtaining a witness threshold costs a separate branch call. Here e_D is the numerical half-width returned by the exact reference. The defaults are $\gamma = e_D = 10^{-8}$. For linear models, $L_* = \max_i \|x_i\|^2 + 0.03$ controls the initial step $1.5/L_*$; x_i here denotes observation i's input vector. The relaxed solve returns a gradient g_H and common threshold t_H, and a stored risk interval is called tight when its width is at most $2.1e_D$. Section C.10 gives the reference error guarantee.

Algorithm 3 Training controller and all global-reference calls

Require: Initial θ, method, decrease margin γ, reference allowance e_D, at most 100 training positions

Start the policy timer; build the model and native solver; $(L_0, U_0, B_*) \leftarrow \mathsf{Ref}(\theta)$; mark the baseline tight
Initialize an empty four-branch threshold cache; for a shared-threshold method, reevaluate the losses, query B_* at θ, and cache its threshold
for each training position before stopping **do**
  Solve the relaxation once at θ for (g_H, t_H); set $\eta = 1.5/L_*$
  for at most 12 backtracking trials **do**
    Set $y = \theta - \eta g_H$
    if the method is direct DP **then**
      $(L_1, U_1, B_1) \leftarrow \mathsf{Ref}(y)$; accept if $U_1 < L_0 - \gamma$, otherwise reject this trial
    else
      Set T_0 to t_H and the cached thresholds; set $b = 0$ for probe/root or $b = 64$ for probe/tree
      Run Algorithm 1 at $(y, L_0 - \gamma, U_0)$; update the cache from its entire query trace
      if the returned comparison is unknown **then**
        $(\ell, u, B) \leftarrow \mathsf{Ref}(y)$; charge one endpoint fallback
        Reevaluate the losses at y, query branch B for its threshold, and update the cache; charge the extra branch call
        Intersect endpoint information: $L_1 \leftarrow \max\{L_1, \ell\}$, $U_1 \leftarrow \min\{U_1, u\}$; retain the larger lower-bound witness
        Reclassify using (L_0, U_0, γ)
      end if
      if the comparison remains unknown and the baseline is loose **then**
        $(L_0, U_0, B_*) \leftarrow \mathsf{Ref}(\theta)$; charge one baseline refresh, mark tight, and reclassify the existing endpoint
      end if
    end if
    if the endpoint is accepted **then**
      Set B_* to the endpoint's lower-bound witness when available
      If L_1 is not finite, reevaluate the losses and set $L_1 \leftarrow \widehat{f}_{B_*}(y) - e_D$; charge this branch call establishing the new lower bound
      Set $\theta \leftarrow y$ and $(L_0, U_0) \leftarrow (L_1, U_1)$; mark tight exactly when $U_1 - L_1 \leq 2.1 e_D$; end backtracking
    else
      Set $\eta \leftarrow \eta/2$
    end if
  end for
  Stop training if all 12 trials failed
end for

A cache entry is keyed by the sorted group identifiers defining its retained set. Repeated queries move it to the newest position; a fifth distinct entry evicts the oldest. Rejected queries persist. Only thresholds transfer between endpoints, where all responses and priorities are recomputed using the continuing policy random stream. Fallback and witness-threshold calls each perform their own charged forward evaluation.

Runtime includes both successful and rejected trials, initialization, model evaluation and differentiation, branch and tree bounds, and every exact solve. We also count branch queries, node expansions,

assignments, and DP calls to explain the observed savings. Independent accuracy checks take place after training and cannot influence its decisions.

C.4.1 DETAILED TREE SEARCH

Algorithm 1 gives the decision sequence. The following implementation specifies how to order nodes without listing all feasible branches. A node represents a set of retained collections; its descendant count is the number of feasible leaves below it. Greedy search orders nodes by their stored upper bounds. For uniform search, an exponential random variable with rate equal to the descendant count gives each leaf the same chance of being selected first. Here $\mathrm{Exp}(n)$ denotes an exponential distribution with rate n and mean $1/n$, p_N is node N's priority, and n_1, n_2 count the leaves below its children. The split index J chooses which child inherits the parent's priority. Ties follow insertion order.

The frontier is the set of nodes still awaiting processing. The algorithm stores the largest upper bound among closed nodes, U_{closed}, and the largest queried-leaf lower bound, L_{leaf}. The expansion count E is limited by b; leaf queries do not increment E. The bound u_N is accompanied by a count of the thresholds it already uses. Adding a threshold preserves the validity of all older upper bounds, so a node needs updating only when it is processed. The probe's lower bound is retained separately. These rules make the budget affect work, while every returned decision still follows the interval tests.

Algorithm 4 Detailed tree search for endpoint checking

Require: Endpoint y, comparison $c = L_0 - \gamma$, baseline U_0, relaxed/cached thresholds T, stored witness B_* from the fixed probe rule, expansion budget b

Evaluate losses and check numerical allowances; return unknown on a failed range check; query B_*'s enclosed risk and fixed quantile; return its witness if the lower bound is at least U_0
Add its quantile to T; build each distinct group-score matrix once; compute the root bound and pool version; return descent with the root enclosure if its bound is below c
Set $U_{\mathrm{closed}} = L_{\mathrm{leaf}} = -\infty$ and expansions $E = 0$; queue the root with priority $p \sim \mathrm{Exp}(n_{\mathrm{root}})$ for uniform search, or key $-u_{\mathrm{root}}$ for greedy search
while the frontier is nonempty and $E < b$ **do**
    Pop the smallest key; if its stored bound is at least c, process new thresholds and refresh its bound and version
    if the bound is below c **then**
        Close the node, retaining its upper bound in U_{closed}
    else if the node is a leaf **then**
        Query its enclosed risk and fixed quantile; update the largest leaf lower bound and closed upper bound; return a witness if its lower bound is at least U_0
        Add a new quantile to T, building its scores once; keep queued priorities fixed
    else
        Split and normalize children of counts n_1, n_2; charge $E \leftarrow E + 1$; compute each child's bound using all current thresholds
        Uniform: choose J with probability $n_J/(n_1 + n_2)$, inherit $p_J = p_N$, and draw $p_{3-J} = p_N + \mathrm{Exp}(n_{3-J})$; greedy: use each child's insertion bound; queue both children
    end if
end while
Return $U_1 = \max\{U_{\mathrm{closed}}, \max_{N \in \mathrm{frontier}} u_N\}$ and the largest probe/leaf lower bound; accept if $U_1 < c$, otherwise return unknown

C.5 A COMPLETE MATCHING ENDPOINT

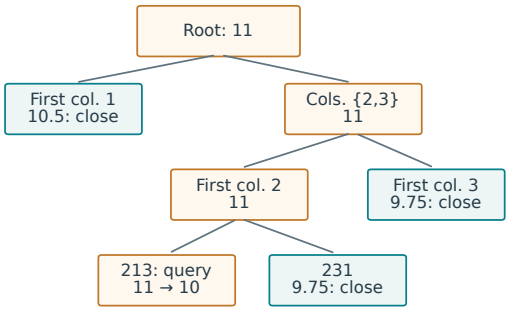
(a) Threshold pool and candidates

Branch	{0}	{0, 9}	{0, 9, 5}
123	**12.75**	10.5	10.5
132	8.75	8.75	8.5
213	**11**	**11**	10
231	9.75	9.75	8.75
312	9.5	9.5	8
321	**12.25**	9.25	9.25

Candidates: 3 ⟶ 1 ⟶ 0

Probe 123: $t = 9$; query 213: $t = 5$

(b) Completion bounds and pruning



$b = 0$: unknown → DP; full tree: $U_1 = 10.5$

Figure 4: Executing the threshold-cover mechanism in Figure 1. The probe adds threshold 9, reducing the unresolved leaves from three to one. Querying branch 213 adds threshold 5 and completes the cover. The matching tree closes subfamilies through completion bounds; zero expansion budget instead invokes DP at the unresolved root.

This example shows why multiple shared thresholds can prove descent when every single-threshold test fails, and how tree splitting makes that information usable. The example in Figure 1 uses a 3×3 grid, two observations per group, tail mass $\beta = 2/3$, and $R = 0$. At the proposed endpoint, the loss pairs are

$$\begin{pmatrix} (0,9) & (12,2) & (8,7) \\ (2,5) & (9,10) & (2,8) \\ (8,7) & (4,12) & (12,11) \end{pmatrix}.$$

A branch retains one cell per row and column, so its CVaR is the average of its four largest losses. Its identifier lists the retained column in each row. Set $L_0 = U_0 = c = 85/8 = 10.625$ and $\gamma = 0$. A compatible current point gives diagonal cells loss pair $(0, 85/6)$ and other cells $(0, 1)$. Branch 123 uniquely maximizes its risk at $85/8$; its canonical threshold and the relaxed threshold are zero. The following exact values trace the threshold pool; an asterisk marks an upper bound at least c.

Branch	f_B	t_B	$u_B(\{0\})$	$u_B(\{0,9\})$	$u_B(\{0,9,5\})$
123	10.5	9	12.75^*	10.5	10.5
132	8.25	4	8.75	8.75	8.5
213	10	5	11^*	11^*	10
231	8.75	7	9.75	9.75	8.75
312	8	5	9.5	9.5	8
321	8.75	8	12.25^*	9.25	9.25

The initial pool is $\{0\}$ and the stored feasible witness is 123. Its probe returns $f_{123} = 10.5$ and $t_{123} = 9$. For the augmented pool $T_0 = \{0, 9\}$, the only candidate is 213: $K = P_{\mathcal{C}} = 1$. Retaining $t_0 = 0$ gives $K_0 = 3$. Across the full family, $P = 2$: the pool $\{5, 9\}$ resolves every branch, whereas the best single threshold gives $V_{\mathrm{H}} = 43/4 = 10.75 > c$. At $t = 8$, branches 123 and 213 both have response $43/4$ and slopes $-1/4$ and $1/4$. Their equal mixture is minimized there, proving the matching lower bound on V_{H}.

A bound for the whole tree differs from the maximum of the separate branch bounds: it must use one threshold for all descendants at a time. The root bound is $\min_{t \in \{0,9\}} \max_B F_B(y, t) = 11$. The first binary split places first-row column 1 against columns $\{2, 3\}$. Their bounds are 10.5 and 11, so the first child closes. Splitting the second child gives first-row columns 2 and 3, with bounds 11 and 9.75; column 3 closes. The column-2 child then splits into leaves 213 and 231, whose bounds are 11 and 9.75. Querying 213 returns value 10 and threshold 5. The closed-node upper bound is now 10.5 and the frontier is covered, establishing descent. Greedy search uses three internal expansions and one additional leaf query after the probe. A zero-expansion check returns unknown at the root and invokes the exact reference, which gives $V_{\mathrm{D}} = 10.5$. The probe sets $L_1 = 10.5$; tree completion

lowers U_1 from 11 to 10.5. If every final shared upper bound is enlarged by at most ϵ, the same pool still resolves descent whenever $\epsilon < 1/8$, the available excess decrease in this example. All arithmetic here is rational, separating the sequence of operations from the numerical allowances used in the measured experiments.

C.6 Numerical and statistical reporting

Calculations use double-precision floating-point arithmetic (binary64) in NumPy and Gurobi 13.0.1. Branch and threshold evaluations carry allowance $2 \times 10^{-11}(1 + \max_i \ell_i + |R|)$, increased to the requested precision-study radius; matched-proposal baselines use $2 \times 10^{-10}(1 + |V_{\mathrm{D}}|)$. Increasing-update counts use 10^{-8}, with raw differences retained. The data-dependent bounds below justify these enclosures.

We average configurations within each independent seed before forming two-sided 95% Student intervals. Each interval is the mean plus or minus a Student t critical value times the standard error across seed means; reported $\pm$ values are interval half-widths. Critical values are 2.093, 2.262, and 2.306 for 20, ten, and nine seeds. Geometry cover sizes one, two, and three have 20, 20, and nine contributing seeds. Runtime intervals are nonnegative. Computation uses an Intel Xeon Platinum 8469C CPU with one numerical and solver thread. Offline reference work is recorded separately from policy time.

C.7 A direct optimum reference for the learning objective

To assess remaining training error, we also compute bounds on the globally optimal linear-model risk. CVaR duality represents each branch risk by nonnegative observation weights q_i, restricted to that branch and capped by the tail mass. Let Q_B be their feasible set and g_q the resulting weighted regression loss plus ridge penalty, with $\lambda = 0.03$. Then $V_{\mathrm{D}}(\theta) = \max_{q \in \cup_B Q_B} g_q(\theta)$, where

$$Q_B = \{q \geq 0 : \mathbf{1}^\top q = 1,\ q_i = 0\ (i \notin B),\ q_i \leq 1/(\beta m)\},$$
$$g_q(\theta) = \tfrac{1}{2} \sum_i q_i (x_i^\top \theta - y_i)^2 + \tfrac{\lambda}{2} \|\theta\|^2,$$

To obtain a lower bound without all feasible weights, keep J weight vectors $q^1, \ldots, q^J$. The auxiliary variable v upper-bounds their unregularized weighted losses in the following smaller optimization problem:

$$\min_{\theta, v}\ v + \tfrac{\lambda}{2}\|\theta\|^2 \quad \text{s.t.} \quad \tfrac{1}{2} \sum_i q_i^j (x_i^\top \theta - y_i)^2 \leq v \quad (j = 1, \ldots, J).$$

Enumerate branch risks at each candidate and add the three highest branches' tail weights, omitting duplicates. Completed runs use seven master variables and at most 30 cuts. Normalize the nonnegative master multipliers into a convex mixture $\bar{q}$. Let X have observation inputs as its rows and y collect their responses. Since $g_{\bar{q}} \leq V_{\mathrm{D}}$, writing $A = X^\top \operatorname{diag}(\bar{q}) X + \lambda I$ and $b = X^\top \operatorname{diag}(\bar{q}) y$ yields

$$\min_\theta g_{\bar{q}}(\theta) = \tfrac{1}{2} \sum_i \bar{q}_i y_i^2 - \tfrac{1}{2} b^\top A^{-1} b \leq \min_\theta V_{\mathrm{D}}(\theta). \tag{37}$$

Enumeration supplies the upper bound. The solver targets gap 10^{-7} and accepts an unchanged cut set below 10^{-6}; the final candidate gives the upper bound and the weight mixture gives the lower bound.

For all 240 configurations, rational arithmetic treats the finite-precision inputs exactly, using feasible tail weights and exactly normalized mixtures. To verify a lower bound without relying on a numerical matrix inverse, use the objective value and gradient at the candidate parameter z. Strong convexity with coefficient λ gives

$$g_{\bar{q}}(z) - \frac{\|\nabla g_{\bar{q}}(z)\|^2}{2\lambda} \leq \min_\theta g_{\bar{q}}(\theta), \tag{38}$$

by minimizing $g_{\bar{q}}(z) + \nabla g_{\bar{q}}(z)^\top (\theta - z) + \lambda \|\theta - z\|^2/2$. Exact fractions evaluate this lower bound and every sorted fractional branch tail at z. Outward rounding to a 10^{-10} grid gives optimum intervals of maximum width 1.862×10^{-7} and mean width 5.72×10^{-9}. Figure 25 uses the lower endpoint. Direct-reference time includes master solves, separation, and floating-point bounds; rational verification is recorded separately.

C.8 FORWARD-ERROR BOUNDS FOR STORED FEATURE DESIGNS

The same dimension-dependent calculation covers the original covariates and fixed nonlinear feature designs. Treat the finite-precision design matrix, targets, and parameters as fixed exact inputs. This separates arithmetic error during risk evaluation from statistical error or feature approximation. The notation fl denotes the stated floating-point evaluation and hats mark its computed values. Use binary64 round-to-nearest arithmetic with finite normal intermediates, allowing exact zeros, and let $u = 2^{-53}$ and $\gamma_j = ju/(1-ju)$ (Higham, 2002). The unit roundoff is u, and γ_j bounds accumulated relative error over j operations. For d features, computed residual $\widehat{r}_i$, half-squared loss $\widehat{\ell}_i$, and regularizer $\widehat{R}$, first bound the residual error by $e_{r,i}$ and the regularizer error by e_R. Their scale $\widehat{A}_i$ is the sum of absolute input contributions:

$$\widehat{A}_i = \mathrm{fl}\left(\sum_{j=1}^{d} |X_{ij}\theta_j| + |y_i|\right), \quad e_{r,i} = \gamma_{4d+8}\widehat{A}_i, \quad e_R = \gamma_{4d+8}|\widehat{R}|.$$

The residual dot product and subtraction have error at most γ_{d+2} times the exact absolute scale. Computing that nonnegative scale adds a factor bounded by $1/(1-\gamma_{d+1})$; the coefficient $4d+8$ covers both. The regularizer's nonnegative square sum obeys the same bound, including representation of the stated ridge coefficient. Expanding the squared residual gives

$$e_{\ell,i} = |\widehat{r}_i|e_{r,i} + \tfrac{1}{2}e_{r,i}^2 + \tfrac{\gamma_4}{2}\widehat{r}_i^2.$$

The first two terms bound perturbation of the exact square and the last its floating-point evaluation. For a retained size m among G groups, use operation-depth allowance $N = 8m+8G+64$ and maximum computed loss $M = \max_i \widehat{\ell}_i$. In this local error calculation, T is a scalar magnitude bound on M and every supplied threshold, rather than the threshold set used elsewhere. Since subsequent canonical thresholds are observed losses, the same T covers discovery. A shared response satisfies

$$E_F(N) = \frac{\max_i e_{\ell,i}}{\beta} + e_R + \gamma_N\left[1 + T + \frac{M+T}{\beta} + |\widehat{R}|\right]. \tag{39}$$

The hinge is one-Lipschitz in each loss, giving the first term after summing m observations and dividing by βm. The conservative depth N covers hinge subtraction, group sums, retained set aggregation, the denominator, and final additions. For a directly evaluated branch minimum, unit mass of the CVaR risk envelope and its nonnegative tail sum instead give

$$E_B = \max_i e_{\ell,i} + e_R + \gamma_{m+16}(1 + M + |\widehat{R}|).$$

This shorter count applies to partial-selection branch queries; streaming enumeration retains its recorded looser count $8m+128$. Sorting and maximization are one-Lipschitz in the uniform norm and amplify no error. To include rounding in the error bound itself, use twice the computed positive $E_F(N)$ plus 2^{-1022}. The added term is the smallest positive normal binary64 value and covers the transition to subnormal numbers near zero. Sections C.9 and C.10 prove the additional completion and DP bounds.

Error bounds on the evaluated designs. The original $d = 6$ proposal grid used the looser constants 512 for responses and 256 for branch sums. Its 6,480 endpoints have maximum response bound 4.7355×10^{-11}, maximum bound/allowance ratio 0.014923, and maximum baseline bound 3.293×10^{-13}. The larger $d = 32, m = 96$ study uses N above: its response bound reaches 3.3774×10^{-12} and ratio 0.02635. Streaming enumeration's maximum error bound is 1.0794×10^{-12}, with minimum remaining sign margin 2.5117×10^{-6}. The formulas apply to other stored feature dimensions through the same objective enclosure.

Signed sweeps use exact integer counts. Their slope error is at most $\gamma_5(1 + 1/\beta)$; the additional $10^{-12}(1+\text{bracket width})$ allowance covers its supporting-line contribution on the primary grid. The brackets may be interpreted for rounded losses, with Equation (39) transferring their values to the exact stored-input objective through tail-order changes.

C.9 NUMERICAL UPPER BOUNDS FROM ROUNDED INTEGER SCORES

Tree and leaf bounds must use the same numerical response to preserve the uniform-query analysis. We achieve this by rounding group scores upward onto a fixed grid and solving the resulting

assignments with exact integers. This section bounds the additional error, using the forward-error conditions of Section C.8. Stored binary64 inputs are interpreted as exact real inputs. Let $u = 2^{-53}$, $\gamma_r = ru/(1-ru)$, and let fl denote the stated sequence of correctly rounded binary64 operations. Write $k = n \geq 1$, $s \geq 1$, $0 < \beta \leq 1$, $m = ks$, and $\alpha = \beta m$ for the real normalizer. To evaluate the tail normalizer using binary64 arithmetic, let $\widehat{c}$ be its rounded value:

$$\widehat{c} = \mathrm{fl}(\mathrm{fl}(\beta k)s).$$

Let $\widehat{\ell}_i$ and $\widehat{R}$ be the computed endpoint losses and regularizer, $M = \max_i \widehat{\ell}_i$, and $T = \max\{M, \max_{t \in T_0} |t|\}$. Every subsequent branch threshold is a computed endpoint loss and therefore has magnitude at most T.

Error before rounding scores to integers. The computed group score sums the positive loss excesses at threshold t. For a group g, let

$$\widehat{a}_g(t) = \mathrm{fl}\left(\sum_{i \in g} [\,\mathrm{fl}(\widehat{\ell}_i - t)\,]_+\right).$$

The check in Section C.8, with $N = 8m + 8G + 64$, supplies a uniform bound E_g for

$$\left| t + \frac{\sum_{g \in B} \widehat{a}_g(t)}{\alpha} + \widehat{R} - F_B(t) \right| \leq E_g \qquad (|t| \leq T). \tag{40}$$

Here the sum in Equation (40) is interpreted over the computed group scores in exact real arithmetic. To see why the same budget covers this expression, loss formation contributes at most $\max_i e_{\ell,i}/\beta$, regularization contributes e_R, and threshold subtraction and group sums contribute at most $\gamma_N C_{\mathrm{old}}$, where

$$C_{\mathrm{old}} = 1 + T + |\widehat{R}| + (M+T)/\beta.$$

These are components of the existing forward check; its additional aggregation and final-evaluation allowances only enlarge the bound. The implementation doubles their computed nonnegative sum before testing the returned bound.

To choose a score grid and a safe final error allowance, let e_0 be the nominal response allowance and δ the largest power of two no greater than it. The scale C_* bounds the magnitudes used in the final arithmetic, E_0 bounds that arithmetic's rounding error, and e combines it with the previously derived score error E_g:

$$e_0 = \mathrm{fl}\Big(2 \cdot 10^{-11}(1 + M + |\widehat{R}|)\Big), \qquad \delta = 2^{\lfloor \log_2 e_0 \rfloor}, \tag{41}$$

$$C_* = 1 + T + |\widehat{R}| + (M + T + \delta/s)/\beta, \qquad \widehat{C} = \mathrm{fl}(C_*),$$

$$E_0 = \mathrm{fl}(4\widehat{\gamma}_{32}\widehat{C}), \qquad e = \max\{e_0, \mathrm{fl}(8(E_g + E_0))\}, \tag{42}$$

where $\widehat{\gamma}_{32}$ is the computed γ_{32}. The endpoint passes the forward check only if $E_g \leq e_0$. All these quantities and the grid are fixed before discovery.

The implementation also checks that $\mathrm{fl}(\beta k)$ and $\widehat{c}$ are positive normal numbers and $ks \leq 2^{53}$. It checks finite upper scales for $k(W + \delta)$ and $\widehat{C} + 2e$, each at most one quarter of the largest finite binary64 number, where

$$W = \mathrm{fl}(1.01s(M+T) + 1).$$

These checks keep the group-score and complete-numerator calculations, as well as the final evaluation, in range. Every complete integer score sum is restricted to be below 2^{48}.

Lemma 23 (Quantized matching envelope)**.** *Under these checks and the inherited forward-error conditions, let $Q_g(t)$ be group g's upward-rounded integer score and $Z_B(t)$ their sum for branch B:*

$$Q_g(t) = \lceil \widehat{a}_g(t)/\delta \rceil, \qquad Z_B(t) = \sum_{g \in B} Q_g(t),$$

and evaluate, in this order,

$$U_B(t) = \mathrm{fl}\Big(t + \mathrm{fl}(\delta Z_B(t)/\widehat{c}) + \widehat{R} + 2e\Big). \tag{43}$$

The product $\delta Z_B(t)$ is exact, and the additions in Equation (43) are sequential. Then, for all queried thresholds and feasible branches,

$$F_B(t) \le U_B(t) \le F_B(t) + \epsilon, \qquad \epsilon = 3e + \frac{k\delta}{\alpha}. \tag{44}$$

Proof. We first record conservative bounds that account for rounding in the budget itself. All terms defining C_* are nonnegative, and their operation depth is below eight. The scale $C_* \ge 1$ and the normal-range conditions give

$$\tfrac{3}{4}C_* \le \widehat{C} \le \tfrac{5}{4}C_*, \qquad 2\gamma_{32}C_* \le E_0 \le 8\gamma_{32}C_*.$$

These loose factors also cover rounding of $\widehat{\gamma}_{32}$. For the base allowance, $e_0 \le 6 \cdot 10^{-11}C_*$. Since $E_g \le e_0$, the formula for e implies

$$E_g + E_0 \le e/7, \qquad e \le C_*. \tag{45}$$

Indeed rounding $E_g + E_0$ down changes it by at most a factor $1 - u$, and the multiplication by eight is exact in range, so $e \ge 8(1-u)(E_g+E_0) \ge 7(E_g+E_0)$. Conversely, $e \le 16(e_0+E_0) < 10^{-8}C_*$ in binary64. The multiplication $2e$ is likewise exact.

The positive summation term in the doubled forward check gives $E_g \ge \gamma_N C_{\mathrm{old}}$; the factor two absorbs the short rounding paths in that term. Together with $E_g \le e_0$ and $e_0 \le 6 \cdot 10^{-11}C_{\mathrm{old}}$, this gives $\gamma_s \le \gamma_N \le 6 \cdot 10^{-11}$. For the computed losses, subtraction followed by nonnegative summation therefore yields

$$0 \le \widehat{a}_g(t) \le (1+\gamma_s)s(M+T) \le 1.01s(M+T).$$

The slack in W also covers its own rounding. Thus the checked numerator scale bounds all group-score and complete-sum intermediates. A subnormal result of adding or subtracting binary64 numbers is exact; the inherited loss-formation check handles the preceding model evaluation.

Because δ is a power of two, scaling $\widehat{a}_g(t)/\delta$ is exact whenever its nonzero result is normal. A positive subnormal quotient is below one and has ceiling one. The implementation also maps a positive score whose scaled quotient underflows to zero to the integer one. Thus its integer conversion computes the mathematical ceiling, and

$$0 \le d_B(t) := \delta Z_B(t) - \sum_{g \in B} \widehat{a}_g(t) < k\delta. \tag{46}$$

The range restriction gives exact conversion of $Z_B < 2^{48}$ to binary64. Since e_0 is bounded below by a positive constant near $2 \cdot 10^{-11}$, δZ_B is either zero or normal. Its power-of-two scaling preserves the significand, and the numerator range condition excludes overflow. The product is therefore exact. A positive quotient by $\widehat{c}$ is also normal, since $\widehat{c} \le 2^{53}$ and δ is bounded below as above.

The two normalizer multiplications satisfy

$$\widehat{c} = \alpha(1+\vartheta), \quad |\vartheta| \le \gamma_2, \qquad |\alpha/\widehat{c} - 1| \le \gamma_4.$$

Set $q = \delta Z_B(t)$. From the group-score bound and Equation (46),

$$q/\alpha \le (1+\gamma_s)(M+T)/\beta + \delta/(\beta s) \le 1.01C_* < 2C_*.$$

Consequently replacing α by $\widehat{c}$ has error at most $2\gamma_4 C_*$.

Following the exact product, the remaining evaluation has one division and three additions. Its error relative to $t + q/\widehat{c} + \widehat{R} + 2e$ is at most

$$\gamma_4\big(T + q/\widehat{c} + |\widehat{R}| + 2e\big).$$

The checked evaluation scale, together with $q/\widehat{c} \le 1.02C_*$, keeps these operations finite. Using $T + |\widehat{R}| \le C_*$, $e \le C_*$, and the looser $q/\widehat{c} \le 2.01C_*$, the total normalizer and evaluation error is less than

$$8\gamma_4 C_* \le 2\gamma_{32}C_* \le E_0.$$

This includes the rounding of the final padding addition; the formation of $2e$ itself is exact.

Combining this result, Equation (40), and Equation (46) gives

$$\left| U_B(t) - \left(F_B(t) + 2e + \frac{d_B(t)}{\alpha} \right) \right| \le E_g + E_0 \le e/7.$$

The lower endpoint is above $F_B(t)$, and the upper endpoint is at most $F_B(t) + (15/7)e + k\delta/\alpha$. The simpler bound Equation (44) follows. □

Using the same response for nodes and leaves. For fixed t, let $\psi_t(Z)$ be the common evaluation in Equation (43). Correct rounding and positive scaling make it nondecreasing in Z. Exact integer assignment, including the forced-prefix integer sum before evaluation, therefore gives

$$\psi_t\left(\max_{B\in\mathcal{B}_N} Z_B(t)\right) = \max_{B\in\mathcal{B}_N} U_B(t).$$

The assignment routine bounds its row and column potentials (dual variables) and slacks (differences from candidate edge costs) by 2^{56}. Arithmetic skips forbidden assignments; each intermediate operation combines at most three such operands, within the range of signed 64-bit integers. A failed range check leaves the comparison unresolved. Taking the minimum over the shared pool preserves the node domination and exact-leaf conditions of Theorem 9. Its initial candidate and critical sets use these same numerical envelopes.

The quantized envelopes need not be convex. For the initially unresolved family, use the true intervals $J_B = \{t : F_B(t) < c - \epsilon\}$ with ϵ from Equation (44). If the fixed representatives satisfy $F_B(t_B) - f_B \le \zeta$ and $\max_B f_B \le c - \mu$ with $\mu > \epsilon + \zeta$, then every representative lies in its J_B. The harmonic bound of Corollary 7 applies to their piercing number, while the numerical node equality preserves the implicit query law.

C.10 A PATHWISE ERROR BOUND FOR THE EXACT MATCHING REFERENCE

An exact combinatorial DP still evaluates real-valued losses in floating-point arithmetic. We bound this numerical error along each feasible DP path; taking the maximum then gives a global bound without multiplying the error by the number of states. Let M bound all computed losses, $0 \le \widehat{\ell}_i \le M$, let $|\widehat{\ell}_i - \ell_i| \le e_{\ell,i}$ and $|\widehat{R} - R| \le e_R$, and set $\alpha = \beta n s$. We use binary64 round-to-nearest arithmetic, finite reachable intermediates, safe integer indices, and positive normal products in $\widehat{c} = \mathrm{fl}(\mathrm{fl}(\beta n)s)$. These are the evaluation conditions checked in Section C.9. Write $\gamma_r = ru/(1-ru)$, $u = 2^{-53}$, with $(s+n+16)u < 1/2$.

Proposition 24 (Global DP evaluation error)**.** *Let $\widehat{D}$ be the native DP value after division by $\widehat{c}$ and addition of $\widehat{R}$. Then*

$$|\widehat{D} - V_{\mathrm{D}}| \le E_{\mathrm{DP}} := \max_i e_{\ell,i} + e_R + \gamma_{s+n+16}(1 + M + |\widehat{R}|). \tag{47}$$

The bound is independent of the number of DP states.

Proof. Put $j = \lfloor \widehat{c} \rfloor$. The subtraction $\rho = \widehat{c} - j$ is exact: for $j = 0$ it subtracts zero, and for $j \ge 1$ the two operands satisfy Sterbenz's factor-two condition. Thus the fractional flag and $0 \le \rho < 1$ describe the exact capped-simplex vertices at mass $\widehat{c}$. The complementary-count representation preserves these same allocations.

Each complete path contains n group rewards. Each unit-weight observation passes through at most s prefix additions and n path additions; there is at most one fractional multiplication and one additional fractional addition. Set $p = s + n + 3$. For a path with exact numerator V, its computed numerator $\widehat{V}$ therefore satisfies

$$|\widehat{V} - V| \le \gamma_p \widehat{c} M + \eta,$$

where $\eta = 2^{-1074}$ also covers gradual underflow in the unique fractional product. Additions producing subnormal results are exact. The nonnegative observation weights sum to $\widehat{c}$, which accounts for the numerator scale. Each transition is monotone in its predecessor value, so unrolling the recurrence expresses its final value as the maximum of these computed path values. The maximum is one-Lipschitz in the uniform norm. Consequently the same error bound holds for the global numerator, regardless of how many states or alternatives are compared. Sorting and exact floating-point comparisons add no arithmetic error.

Division by $\widehat{c}$ and addition of $\widehat{R}$ enlarge the normal-range error to $\gamma_{p+2}(M + |\widehat{R}|)$. Since $\widehat{c}$ is normal, $\eta/\widehat{c} \le 2u$; allowing both this propagated underflow and final rounding adds at most $5u$.

It remains to compare the rounded and real tail masses, including integer-boundary crossings. For any retained set's decreasing loss quantile $z(v) \in [0, M]$, define $S(a) = \int_0^a z(v)\,dv$. For any

$a, b \in (0, ns]$,

$$\left|\frac{S(a)}{a} - \frac{S(b)}{b}\right| \le M\frac{|a-b|}{\max\{a,b\}}.$$

This follows by splitting the integral at the smaller mass; the added segment's mean and the original tail mean both lie in $[0, M]$. The two multiplications give $|\widehat{c} - \alpha| \le \gamma_2\alpha$. Thus their effect is at most $\gamma_4 M$, a bound that also allows one initial rounding of a specified tail mass. No assumption on equality of the two integer parts is needed. The sum of these arithmetic terms is bounded by $\gamma_{s+n+16}(1 + M + |\widehat{R}|)$. Finally, CVaR and the maximum over retained sets are one-Lipschitz under a uniform loss perturbation, giving the first two terms of Equation (47). □

The existing forward check can be reused directly. Its depth $N = 8ns + 8n^2 + 64$ satisfies $N \ge s + n + 16$, its loss term is $\max_i e_{\ell,i}/\beta \ge \max_i e_{\ell,i}$, and its scale $1 + T + (M + T)/\beta + |\widehat{R}|$ is at least $1 + M + |\widehat{R}|$. Its checked, doubled bound E_g therefore also bounds E_{DP}. For a fixed baseline half-width $r = 10^{-8}$, outward-rounded bound arithmetic can check

$$E_g + u(|\widehat{D}| + r) + 2^{-1022} \le r.$$

This includes rounding of the two interval endpoints. Alternatively, round $\widehat{D} - E_g$ and $\widehat{D} + E_g$ outward.

C.11 Availability of supporting materials

The supplementary results support the runtime, decision-accuracy, and threshold-geometry comparisons reported here. They use the same study configurations and statistical units as the manuscript. Public datasets are available from their cited repositories. The full implementation and detailed experimental records are available from the corresponding author upon reasonable request after anonymous review. During review, requests for additional verification material can be made through the venue's anonymous discussion channel.

D Additional experiments and ablations

We report the public-data and tolerance studies, followed by controlled computational comparisons and ablations.

D.1 Chronological public-data evaluation

This study keeps the same squared-loss CVaR objective, native matching DP, and shared-threshold controller, replacing generated observations by recorded calendar groups. It tests the cost of enforcing discrete descent and measures heldout risk separately across all 360 trajectories and 36,000 iteration positions.

Sources, splits, and calendar support. We use the hourly Bike Sharing and Appliances Energy Prediction datasets from the University of California, Irvine (UCI) Machine Learning Repository, described by Fanaee-T & Gama (2014) and Candanedo et al. (2017), respectively. Both datasets use the Creative Commons Attribution 4.0 license (CC BY 4.0) and are available from the cited UCI repositories. Their preprocessing and chronological splits are specified below.

Bike has 17,379 hourly observations. Calendar year 2011 supplies 8,645 training rows, and all 8,734 rows from 2012 form the heldout population. The 12×12 family uses month by two-hour window; the 6×6 family merges consecutive pairs of months and uses four-hour windows. Their smallest training/heldout cells contain 42/56 and 204/235 rows, respectively. Energy has 19,735 observations at ten-minute intervals. We split at 2016-04-15 00:00:00, giving 13,578 earlier training rows and 6,157 heldout rows. Its 6×6 family uses Monday–Saturday by four-hour window, with minimum training/heldout cell counts 312/144. The 1,872 training and 864 heldout Sunday rows lie outside this six-day matching family; Sundays remain in the full heldout metrics. Timestamps are unique and strictly chronological in both datasets.

Dataset/grid	β	Exact DP	Probe + DP	Greedy + DP	Uniform + DP
Bike 6×6	.75	0.080 ± 0.023	0.118 ± 0.036	0.214 ± 0.092	0.258 ± 0.133
	.90	0.087 ± 0.020	0.133 ± 0.040	0.263 ± 0.120	0.292 ± 0.140
Bike 12×12	.75	3.042 ± 0.905	1.851 ± 0.950	1.518 ± 0.733	2.024 ± 0.989
	.90	1.702 ± 0.350	1.163 ± 0.474	1.152 ± 0.469	1.528 ± 0.645
Energy 6×6	.75	0.078 ± 0.015	0.121 ± 0.028	0.226 ± 0.052	0.256 ± 0.061
	.90	0.114 ± 0.019	0.185 ± 0.035	0.390 ± 0.087	0.437 ± 0.099

Table 2: Complete policy time in seconds on real calendar groups. Means and 95% Student intervals use ten training-subsampling seeds, conditional on the fixed data and heldout sample. The four policies have identical parameter paths.

Predictors and training-only transformations. Bike predicts total rental count. We exclude the separate casual-user and registered-user counts because they sum to the target, along with the source row index, raw date, and year. Its predictors are temperature, apparent temperature, humidity, windspeed, holiday and working-day flags, fixed indicators for season (levels 1–4), weather (1–4), and recorded weekday (0–6), and sine/cosine encodings of hour and month. An intercept gives 26 parameters. Energy predicts appliance energy consumption in watt-hours from lighting consumption, the nine temperature and nine humidity measurements, and weather-station temperature, pressure, relative humidity, windspeed, visibility, and dew point. It adds sine/cosine time of day and seven calendar-weekday indicators; an intercept gives 35 parameters. The two randomly generated nuisance predictors and the raw timestamp are excluded.

For each dataset, non-intercept predictors are centered and divided by their population standard deviations on the full chronological training split only. A standard deviation below 10^{-12} is replaced by one. For the observed response y_i, let μ_{tr} and σ_{tr} be the training mean and standard deviation of $\log(1+y_i)$. The transformed target is $z_i = (\log(1+y_i) - \mu_{\mathrm{tr}})/\sigma_{\mathrm{tr}}$, with $\mu_{\mathrm{tr}}, \sigma_{\mathrm{tr}}$ also fitted only on training rows. The initial parameter is zero, including the intercept. The unchanged training objective uses $\ell_i(\theta) = \frac{1}{2}(x_i^\top\theta - z_i)^2$ and $R(\theta) = 0.015\|\theta\|^2$; the intercept is regularized as well.

Sampling and fixed evaluation samples. The ten evaluation seeds are 2100–2109; development uses seed 2099. For every training calendar cell, sample twelve rows uniformly without replacement and preserve their chronological order. Sampling is independent across dataset, grid, and calendar-cell combinations within each seed. The same selection serves both $\beta \in \{0.75, 0.9\}$ and all methods. A balanced heldout sample is fixed independently of outcomes: in each chronologically ordered heldout cell of size N, take positions $\lfloor (j+\frac{1}{2})N/12 \rfloor$, $j = 0, \ldots, 11$. This deterministic rule differs from training subsampling and is identical across seeds and policies. Training and heldout row sets are disjoint. Full-test metrics use the entire chronological heldout population.

Controllers and randomization. The four descent-checking policies follow Section C.3 and Algorithm 3, with margin and reference padding 10^{-8}. The relaxed control accepts its first proposal; the hull control uses the same twelve-step halving rule but requires $V_H(\theta+s) < V_H(\theta) - 10^{-8}$. All runs record 100 positions, carrying stopped parameters forward at zero policy cost. Execution order is randomized independently of uniform branch selection. Training time includes all proposal and checking work. Common data preparation and post-training risk evaluation are measured separately, so they do not affect either the accepted updates or the comparison of checking costs.

Complete-time and numerical results. Table 2 reports all six dataset/grid/tail settings. On Bike 12×12, greedy saves 1.037 ± 0.157 seconds against DP and 0.172 ± 0.104 against probe/root. Uniform saves 0.595 ± 0.324 against DP but adds 0.269 ± 0.156 against probe/root. All 180 paired comparisons between a sharing policy and direct DP have identical saved parameter paths. The four policies accept 20,772 updates, all of which decrease the training objective; the largest enclosed change upper bound is -4.5804×10^{-8}.

Heldout metrics and acceptance controls. Heldout mean squared error (MSE) averages the squared difference between prediction and standardized log target over every heldout observation. Heldout worst-case CVaR uses the full squared residual, without ridge, on the fixed balanced sample and the same matching family and β. We additionally evaluate ordinary CVaR over the full test popu-

lation, MSE in log-transformed target units, and MSE in original target units. For prediction $x^\top\theta$, the original-scale prediction $\widehat{y}$ uses the inverse transform without clipping: $\widehat{y} = \exp(\sigma_{\mathrm{tr}} x^\top\theta + \mu_{\mathrm{tr}}) - 1$. For direct DP, original-unit heldout MSE is $49{,}604 \pm 8{,}710$ counts^2 on Bike 6×6, $48{,}783 \pm 9{,}030$ counts^2 on Bike 12×12, and $8{,}544 \pm 463$ Wh^2 on Energy.

Dataset/grid	Policy	Full-test MSE	Worst CVaR	Increases
Bike 6×6	Descent check	0.7678 ± 0.1728	1.1862 ± 0.3268	0
	Relaxed	0.7444 ± 0.1842	1.1528 ± 0.3428	416
	Hull	0.7662 ± 0.1735	1.1831 ± 0.3280	117
Bike 12×12	Descent check	0.7429 ± 0.1844	1.2359 ± 0.2974	0
	Relaxed	0.7341 ± 0.1887	1.2209 ± 0.3041	300
	Hull	0.7425 ± 0.1846	1.2354 ± 0.2976	27
Energy 6×6	Descent check	0.7355 ± 0.1014	1.2184 ± 0.1666	0
	Relaxed	0.6924 ± 0.0798	1.1533 ± 0.1518	620
	Hull	0.7325 ± 0.1009	1.2146 ± 0.1673	86

Table 3: Heldout squared-error risks in standardized-log-target units, without ridge. Descent check denotes the identical result of direct DP, probe/root, probe/greedy, and probe/uniform. Risks are means and 95% Student intervals after averaging both tail masses within seed. Increases count accepted training V_{D} changes above 10^{-8} over twenty trajectories of each policy.

Intervals use ten training-subsampling seeds, with configurations paired within seed; tail-specific results remain in the released tables. They describe subsampling variability conditional on these fixed data and test samples. The relaxed control has lower mean heldout risk but admits training-risk increases; hull differences are smaller and their paired intervals contain zero. Figure 5 displays paired differences so shared dataset variation does not obscure this comparison. The paired comparisons quantify the cost of enforcing the exact-checking path while preserving its heldout outcomes.

Separating path differences from later updates. A control may finish with lower risk either because it follows a different path or because it continues after direct DP stops. We separate these effects at the direct-DP stopping position. Let θ_j^E be the direct-DP parameter after iteration position j, with $j = 0$ denoting initialization, and let θ_j^C denote a control. If direct DP first halts after failed zero-based iteration i, set $k = i + 1$; otherwise set $k = 100$. Its repeated-state convention gives $\theta_k^E = \theta_{100}^E$. Let G_C denote the final risk gap, A_C the gap at the common position k, and B_C the control's subsequent risk reduction. We record

$$G_C = V_{\mathrm{D}}(\theta_{100}^E) - V_{\mathrm{D}}(\theta_{100}^C),$$
$$A_C = V_{\mathrm{D}}(\theta_{100}^E) - V_{\mathrm{D}}(\theta_k^C), \qquad B_C = V_{\mathrm{D}}(\theta_k^C) - V_{\mathrm{D}}(\theta_{100}^C), \qquad G_C = A_C + B_C.$$

A_C is the gap at a common position; B_C is the control's later risk reduction. Positive G_C means a lower control terminal training risk.

Dataset/grid	Control	A_C: gap at k	B_C: after k	G_C: final gap
Bike 6×6	Relaxed	8.084 ± 5.518	3.676 ± 2.965	11.761 ± 6.849
	Hull	0.230 ± 0.524	0.721 ± 0.881	0.952 ± 1.238
Bike 12×12	Relaxed	5.131 ± 3.312	0.734 ± 0.858	5.865 ± 3.562
	Hull	0.198 ± 0.491	0.176 ± 0.301	0.374 ± 0.554
Energy 6×6	Relaxed	23.276 ± 5.827	12.311 ± 7.570	35.588 ± 11.614
	Hull	0.161 ± 1.003	3.206 ± 2.268	3.366 ± 2.477

Table 4: Exact training-risk decomposition in units of 10^{-3}, including ridge. Means and 95% intervals use the same ten seed means as Table 3; displayed entries are rounded independently. The identity holds before rounding.

Direct DP halts in 27 of the sixty dataset/grid/seed/tail configurations. The mean k is 86.05, 97.20, and 77.75 for Bike 6×6, Bike 12×12, and Energy. The relaxed control performs a further 13.95, 2.80, and 22.25 accepted positions on average; hull performs 13.50, 2.80, and 16.65. Ninety-six controls already differ from direct DP by k; the other 24 coincide throughout. Accordingly, subsequent updating explains part, but not all, of the final gap. For example, the Energy hull gap is mostly improvement after position k, whereas the Bike 12×12 relaxed gap is mostly present at k.

The identity describes the contributions of different paths and stopping positions. Here V_{D} is convex because CVaR, maximization over retained sets, and the ridge term preserve convexity of the squared losses. Halting records failure of the twelve tested relaxed-direction steps. The comparisons therefore concern finite-budget proposal paths and their continuation.

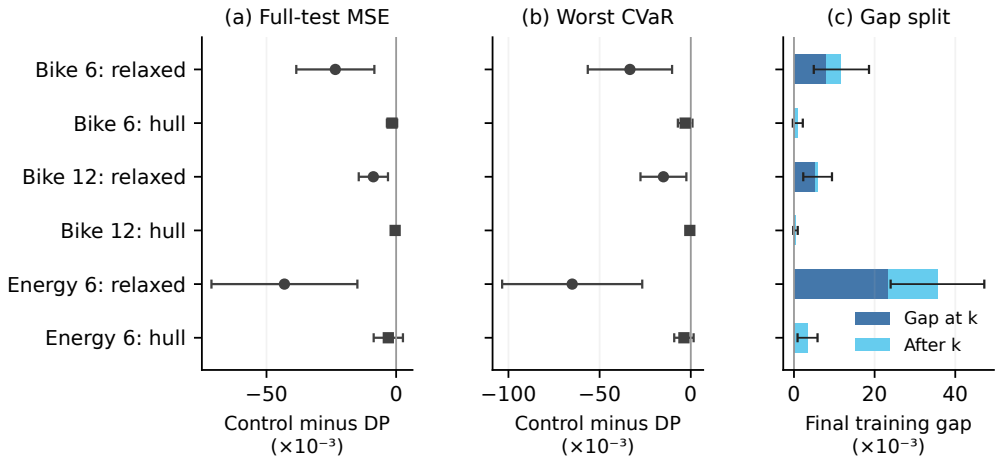


Figure 5: Heldout controls and risk differences at a common stopping position on real data. (a,b) Paired control-minus-direct-DP differences in full-population MSE and balanced-sample worst-case CVaR; negative values favor the control. (c) Mean terminal training-risk gap split into the gap at k and the control's subsequent reduction; error bars show the total gap's 95% interval. All panels display units of 10^{-3} and use beta averages within each of ten seeds. The four descent-checking methods coincide.

Study scope. The estimates condition on two datasets, one model class, fixed chronological test periods, and the stated calendar matching families. A larger Bike grid also supplies more training observations under the twelve-per-cell rule.

D.2 Fixed observations and nonlinear feature models

This comparison varies the prediction features and matching granularity while holding the descent rule fixed. It uses the chronological splits, targets, full heldout populations, and four descent-checking policies of Section D.1, with ten subsampling seeds 3100–3109. Three feature designs, three dataset/grid settings, two tail masses, and four policies give 720 trajectories.

Training-only feature construction. Alongside the original linear predictors, we use fixed random Fourier features (Rahimi & Recht, 2007). Let z contain the training-standardized non-intercept covariates. On a fixed random sample of 512 training observations, set ℓ to the square root of the median positive pairwise squared distance. Draw 512 independent columns $w_j \sim N(0, I)/\ell$ and phases $b_j \sim \mathrm{Unif}[0, 2\pi)$ using independent feature randomness fixed across all comparisons. Here ℓ is the kernel bandwidth: larger values produce lower frequencies and more slowly varying features. The vector w_j is a sampled frequency and b_j its phase offset. For D cosine features, append a constant one for the intercept to obtain

$$\phi_D(z) = \big(\sqrt{2/D}\cos(w_1^\top z + b_1), \ldots, \sqrt{2/D}\cos(w_D^\top z + b_D), 1\big), \qquad D \in \{128, 512\}.$$

They share their first 128 frequency/phase pairs, with the stated dimension-dependent normalization. Bandwidths are 6.91695 for Bike and 7.87292 for Energy. The maps remain fixed across seeds, policies, and grids; optimization trains only their linear readout, initialized at zero with ridge coefficient 0.03 including the intercept. The resulting predictors are nonlinear in the original covariates, while the trained objective remains convex. The resulting finite-precision features define the objective bounded in Section C.8.

Holding the observation set fixed. For each Bike seed, select twelve training rows in every fine month/two-hour cell with independent sampling in each cell. Merging each 2×2 block gives 48 observations per coarse group. Both grids therefore use exactly the same 1,728 training rows; the coarse grid retains 288 observations per retained set and the fine grid 144. Their constraints and retained counts differ. The balanced heldout subsets use the same fine-cell midpoint rule as Section D.1 and the same merging, again preserving their observation set. Energy retains its 6×6 design with twelve observations per group and independent sampling in each cell.

The controller uses at most 100 training iterations and twelve backtracking trials per iteration, a four-branch threshold cache, 64 tree expansions, and margin/reference padding 10^{-8}. Complete policy time includes model construction, forward and gradient evaluation, relaxed solves, shared bounds, search, and exact fallback. Feature preparation takes 0.106 seconds for Bike and 0.085 for Energy for the common bandwidth and projection stage; the 128/512-dimensional cosine transforms take 0.051/0.190 and 0.035/0.135 seconds, respectively. These one-time costs prepare the full training and heldout arrays. Adding them to each policy gives its cost without amortizing feature preparation; paired absolute savings are unchanged, while speed ratios move toward one.

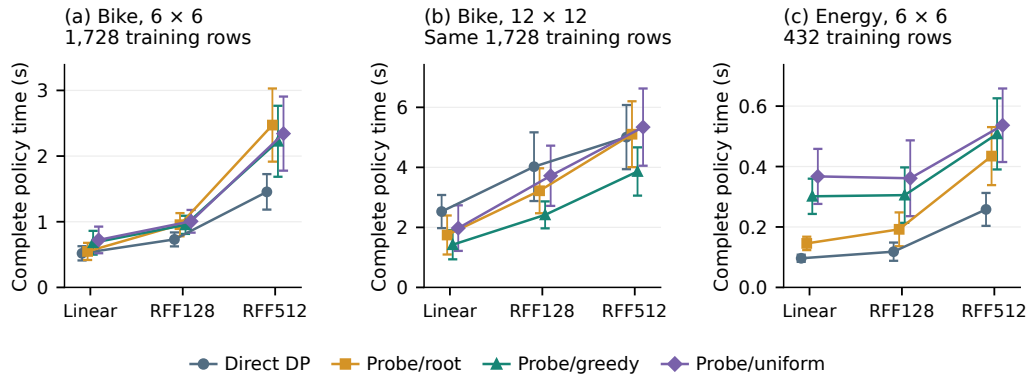


Figure 6: Complete learning cost with fixed observations and nonlinear features. Every point averages the two tail masses within seed, then ten seeds; bars are 95% Student intervals. Bike panels share the training-row set, with group size 48 at 6×6 and 12 at 12×12. Each RFF design trains a linear readout on a fixed nonlinear map. Vertical scales differ across panels.

Cost, path agreement, and heldout risk. Direct DP leads for both small families in every feature design. At Bike 12×12, greedy saves 1.105 ± 0.228, 1.606 ± 0.726, and 1.146 ± 0.317 seconds for linear, RFF128, and RFF512 features, respectively. Their DP/greedy mean-time ratios are 1.78, 1.66, and 1.30. With RFF512, probe/root and uniform exceed DP's mean time, whereas greedy retains a saving. Thus the relative cost of forward evaluation and threshold search also affects the useful regime.

Of 540 search-versus-DP pairs, 525 have identical complete parameter paths. For all pairs, we also compare the longest prefix with identical inputs, outputs, and tested-trial counts. Greedy's Bike 12×12 savings on these prefixes are 1.108 ± 0.228, 1.606 ± 0.831, and 1.124 ± 0.303 seconds, over mean prefix lengths 99.35, 96.75, and 99.65 positions. The largest absolute terminal-risk difference across any pair is 1.054×10^{-4}.

On Bike 12×12, direct DP's full-test standardized-log MSE is 0.69571, 0.81670, and 0.81764 for the three designs; greedy gives 0.69571, 0.81667, and 0.81765. The experiment measures descent-checking cost under fixed features and regularization. Full per-tail costs, common-prefix comparisons, and heldout metrics accompany Table 5.

Data/grid	Features	β	DP	Probe	Greedy	Uniform
Bike 6	Linear	0.75	0.58 ± 0.14	0.57 ± 0.15	0.69 ± 0.20	0.73 ± 0.22
Bike 6	Linear	0.90	0.46 ± 0.10	0.52 ± 0.14	0.67 ± 0.20	0.72 ± 0.22
Bike 6	RFF128	0.75	0.78 ± 0.15	1.00 ± 0.24	0.96 ± 0.18	1.02 ± 0.26
Bike 6	RFF128	0.90	0.68 ± 0.11	0.91 ± 0.17	0.95 ± 0.16	0.99 ± 0.18
Bike 6	RFF512	0.75	1.49 ± 0.54	2.49 ± 0.97	2.17 ± 0.84	2.25 ± 0.89
Bike 6	RFF512	0.90	1.42 ± 0.26	2.45 ± 0.48	2.28 ± 0.56	2.43 ± 0.52
Bike 12	Linear	0.75	3.07 ± 0.77	2.04 ± 0.92	1.50 ± 0.55	2.16 ± 0.92
Bike 12	Linear	0.90	1.98 ± 0.47	1.45 ± 0.50	1.35 ± 0.47	1.79 ± 0.67
Bike 12	RFF128	0.75	4.67 ± 2.06	3.60 ± 1.40	2.43 ± 0.66	3.83 ± 1.54
Bike 12	RFF128	0.90	3.38 ± 0.68	2.84 ± 0.60	2.41 ± 0.46	3.61 ± 0.86
Bike 12	RFF512	0.75	5.98 ± 1.54	5.87 ± 1.65	4.27 ± 1.20	5.98 ± 1.74
Bike 12	RFF512	0.90	4.04 ± 1.10	4.34 ± 1.21	3.45 ± 1.04	4.70 ± 1.58
Energy 6	Linear	0.75	0.09 ± 0.02	0.12 ± 0.03	0.26 ± 0.07	0.30 ± 0.10
Energy 6	Linear	0.90	0.11 ± 0.02	0.17 ± 0.03	0.34 ± 0.08	0.44 ± 0.13
Energy 6	RFF128	0.75	0.11 ± 0.03	0.16 ± 0.06	0.26 ± 0.09	0.31 ± 0.14
Energy 6	RFF128	0.90	0.13 ± 0.03	0.22 ± 0.07	0.35 ± 0.11	0.41 ± 0.14
Energy 6	RFF512	0.75	0.24 ± 0.06	0.40 ± 0.12	0.45 ± 0.11	0.49 ± 0.14
Energy 6	RFF512	0.90	0.28 ± 0.07	0.47 ± 0.11	0.57 ± 0.15	0.59 ± 0.13

Table 5: Complete policy seconds for every feature/data/grid/tail setting. Entries are means and 95% Student intervals over ten seeds. Probe, greedy, and uniform all use the common DP fallback.

D.3 Training nonlinear features

To test whether checking remains useful when feature learning also costs computation, we train a network with one hidden layer of width $h = 16$. For input dimension d and an observation $x \in \mathbb{R}^d$, the matrix $W \in \mathbb{R}^{d \times h}$ and bias $b \in \mathbb{R}^h$ form the hidden layer; $v \in \mathbb{R}^h$ and scalar bias a form the output. With $\mathrm{ReLU}(z) = \max\{z, 0\}$ applied coordinatewise, the prediction is

$$p_\theta(x) = v^\top \mathrm{ReLU}(W^\top x + b) + a, \qquad \theta = (W, b, v, a).$$

All four parameter blocks are trained. The loss is $\ell_i(\theta) = \frac{1}{2}(p_\theta(x_i) - y_i)^2$ and the regularizer is $R(\theta) = 0.015\|\theta\|^2$. Input weights are initialized independently from $N(0, 1/d)$, output weights from $N(0, 0.01/h)$, and biases at zero. The model has 433 parameters on Bike, 577 on Energy, and 289 on the synthetic data. The fixed RFF maps in Section D.2 instead train only the final linear readout.

Data and optimization. The chronological splits, input standardization, log-target transformation, and heldout populations are those in Section D.1. Ten subsampling seeds, 4100–4109, select twelve training rows from each fine calendar cell. Bike uses the same 1,728 observations at both grid sizes, with 48 observations per group at 6×6 and twelve at 12×12. Energy has 432 training observations. The synthetic comparison uses the generator in Section C, 16 input covariates, twelve observations per group, and grids $n = 6, 8, 10, 12$, giving 432, 768, 1,200, and 1,728 observations. This scaling comparison changes both the matching family and the sample size; the Bike comparison holds the sample size fixed.

At each current parameter, the relaxed LP supplies the CVaR observation weights q_i of Section B.4. Backpropagation computes $\sum_i q_i \nabla_\theta \ell_i + 0.03\theta$, using derivative zero for a ReLU at zero. The initial step size is 0.02, with at most twelve backtracking trials. Training stops after a failed line search or 100 positions. The four policies use the same initial parameters, four-branch cache, 10^{-8} decrease margin and reference padding, and 64-expansion budget for greedy and uniform search. Three public-data settings, two tail masses, ten seeds, and four policies give 240 trajectories; the four synthetic sizes give another 320. Policy and configuration orders are shuffled.

Forward arithmetic and endpoint bounds. Nonlinearity changes the loss calculation but leaves the CVaR threshold representation and matching DP unchanged. For stored binary64 inputs and parameters, write $\widehat{z}_i$, $\widehat{h}_i$, and $\widehat{p}_i$ for the computed hidden preactivation, ReLU output, and prediction. Standard dot-product bounds give an elementwise hidden error $e_{z,i}$ and output-affine error $e_{a,i}$. Since ReLU is one-Lipschitz, a prediction-error bound is

$$e_{p,i} = |v|^\top e_{z,i} + e_{a,i}.$$

Adding subtraction roundoff gives a residual-error bound $e_{r,i}$. The half-squared loss also incurs rounding when squaring and scaling, bounded by $e_{\mathrm{square},i}$. If $\widehat{r}_i = \widehat{p}_i - y_i$, combining this error with residual uncertainty gives

$$e_{\ell,i} \le |\widehat{r}_i| e_{r,i} + \tfrac{1}{2} e_{r,i}^2 + e_{\mathrm{square},i}.$$

The implementation evaluates these nonnegative bounds outward, includes subnormal roundoff, and derives the ridge error e_R from the parameter dot product. These quantities replace the linear forward error in Section C.9 and Equation (47). The fixed 10^{-8} DP padding is used only when the derived allowance is at most half that value. Loss evaluation, error-bound computation, and any repeated forward passes are included in runtime.

Runtime accounting. The training timer starts before input selection and model construction and stops after the last training position. Nested exclusive timers separate forward arithmetic and its error bounds; the relaxed LP and backward pass; threshold scores and assignment completions; tree and queue operations; branch evaluations; direct, fallback, initial, and refresh DP calls; and setup and control overhead. Their sum equals the enclosing wall time. Branch-evaluation time includes the feasible witness reevaluation used by DP. Data loading and solver-environment initialization are shared within a four-policy job and recorded separately. Charging that cost to each policy gives a standalone load-plus-training comparison. Heldout evaluation and independent checks of the training objective occur after training and are excluded from its timing. They do not influence the accepted updates.

The tables average the two tail masses within each seed before computing ten-seed means and 95% Student intervals. Paired differences use the same seed and tail mass. We compare complete parameter and proposal paths and, when they differ, the longest matching prefix. Prefix iteration time sums the recorded matching positions and excludes initialization; complete policy time retains initialization.

Measured costs and learning outcomes. The four policies follow identical parameter and proposal paths in all 140 data/grid/seed/tail configurations. Their 55,588 accepted updates satisfy the prescribed decrease margin. Every parameter block changes from its initial value, so the experiment trains the hidden features as well as the output layer.

On Bike 12×12, direct DP, probe/root, greedy, and uniform take 1.560, 0.757, 0.805, and 0.819 seconds. Greedy's paired saving against DP is 0.755 ± 0.239 seconds; against probe/root it is -0.048 ± 0.062 seconds. The corresponding DP/greedy ratio is 1.94, or 1.92 after charging common loading and solver setup to each method. On Bike 6×6, the four means are 0.485, 0.510, 0.544, and 0.546 seconds; on Energy they are 0.150, 0.205, 0.262, and 0.281. Greedy therefore costs 0.060 ± 0.046 and 0.112 ± 0.044 seconds more than DP in the two small public-data settings.

For the larger Bike network, model evaluation and relaxed optimization account for 53.4% of greedy's total time: forward evaluation with error bounds costs 0.165 seconds, and relaxed solves with backpropagation cost 0.265. Further search costs 0.052 seconds for shared scores and assignments and 0.055 for tree operations. It reduces mean fallback calls from probe/root's 11.6 to 9.9, while assignment calls increase from 492.6 to 7,486. Probe/root also makes 9.65 current-risk refreshes per trajectory, compared with 9.9 for greedy. These costs explain why avoiding more exact solves does not make greedy faster than the initial probe in this setting.

The common loading and solver-environment costs are 0.0119, 0.0129, and 0.0279 seconds for Bike 6×6, Bike 12×12, and Energy. Charging these to every policy preserves paired absolute savings. The cost comparison therefore includes the computation needed to start and complete a training run; post-training predictive evaluation is measured separately.

Full-test standardized-log MSE is 0.7738, 0.7538, and 0.8420 in the three public-data settings; balanced worst-case squared-error CVaR is 1.1128, 1.2307, and 1.3263. Each metric is identical across the four checking policies. The network study changes the model and uses fresh seeds, so these values are not a paired architecture comparison with the earlier linear study.

In the synthetic network study, direct DP is fastest at $n = 6$. At $n = 8, 10, 12$, probe/root costs 0.172, 0.240, and 0.332 seconds, compared with DP's 0.191, 0.387, and 1.387. Greedy is within 0.003 seconds of the probe at each size. The larger reduction against DP therefore comes from the probe and root bounds rather than additional tree exploration in these configurations.

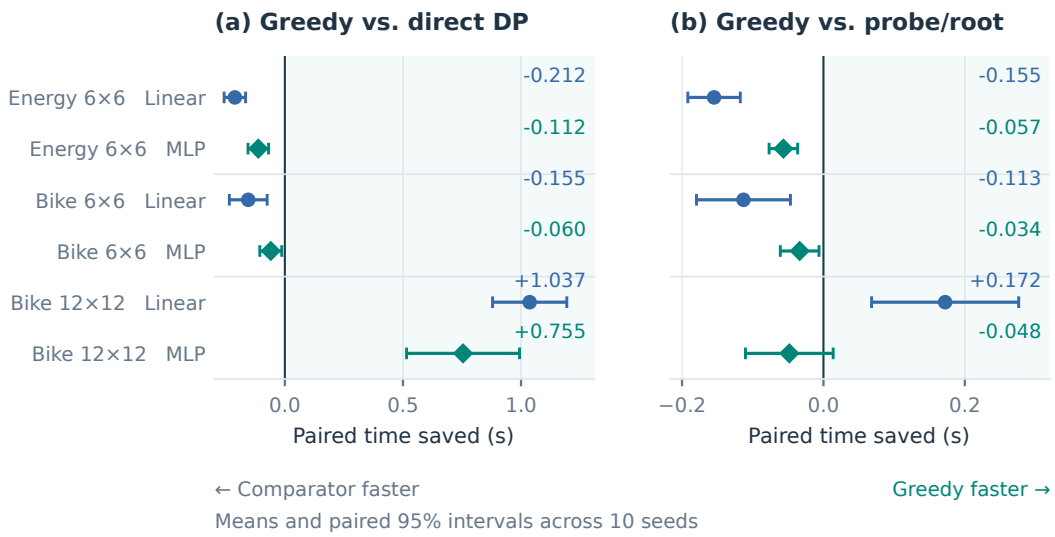


Figure 7: Separating the total benefit of greedy checking from the incremental benefit of search. Positive values denote complete training time saved by greedy, relative to direct DP (left) or probe/root (right). Each contrast pairs policies within the same model and seed, averages the two tail masses, then reports the ten-seed mean and 95% Student interval. Linear and MLP studies use separate seeds. At Bike 12×12, greedy improves on DP for both models; its additional saving over the probe is positive for the linear model, while the MLP interval spans zero.

Data/grid	N	Direct DP	Probe/root	Greedy	Uniform
Synthetic 6×6	432	0.106 ± 0.002	0.114 ± 0.003	0.114 ± 0.003	0.114 ± 0.002
Synthetic 8×8	768	0.191 ± 0.011	0.172 ± 0.009	0.171 ± 0.009	0.174 ± 0.011
Synthetic 10×10	1,200	0.387 ± 0.004	0.240 ± 0.003	0.243 ± 0.005	0.243 ± 0.004
Synthetic 12×12	1,728	1.387 ± 0.011	0.332 ± 0.007	0.333 ± 0.010	0.333 ± 0.008

Table 6: Complete MLP training time in seconds on the synthetic matching families. Each cell is a ten-seed mean and 95% Student interval, after averaging the two tail masses within seed. Group size is twelve.

D.4 Acceptance margins, enclosure width, and solver cost

The comparison studies how the required decrease, numerical uncertainty, and the endpoint solver affect the cost of a decision. A valid enclosure contains the true risk, but it may still overlap the acceptance threshold. Equation (5) separates these two properties: the actual decrease beyond γ must exceed the one-sided uncertainty $e_0 + e_1$. For idealized intervals centered on the exact risks, this uncertainty is $r_0 + r_1$. For general intervals with midpoints $\widehat{V}_0, \widehat{V}_1$ and half-widths r_0, r_1, the observable test is

$$\widehat{V}_0 - \widehat{V}_1 - \gamma > r_0 + r_1.$$

The ratio γ/r alone does not determine acceptance because it omits the proposed decrease. At a fixed proposal, raising γ makes the test harder. Increasing a valid error allowance can leave a truly decreasing proposal unresolved without making any definite decision incorrect.

An unresolved budget-limited search invokes DP. Its endpoint interval may settle the comparison, or a loose current baseline may need refinement. If the true decrease is positive but smaller than γ, neither the acceptance test nor the nondecrease witness is required to succeed, even with exact endpoint values. Unknown therefore denotes insufficient evidence for the specified decision, not an observed risk increase.

Ten fresh data seeds, 420–429, use twelve observations per cell, dimension 32, and tail masses $\beta \in \{0.75, 0.9\}$. The sensitivity grid at $n = 12$ contains six pairs of acceptance margin γ and reference half-width r, shown in Table 7. Four policies give 480 trajectories. A further 160 trajectories compare four reference choices at $n = 8, 12$, with $\gamma = r = 10^{-8}$. Each trajectory has 100 saved

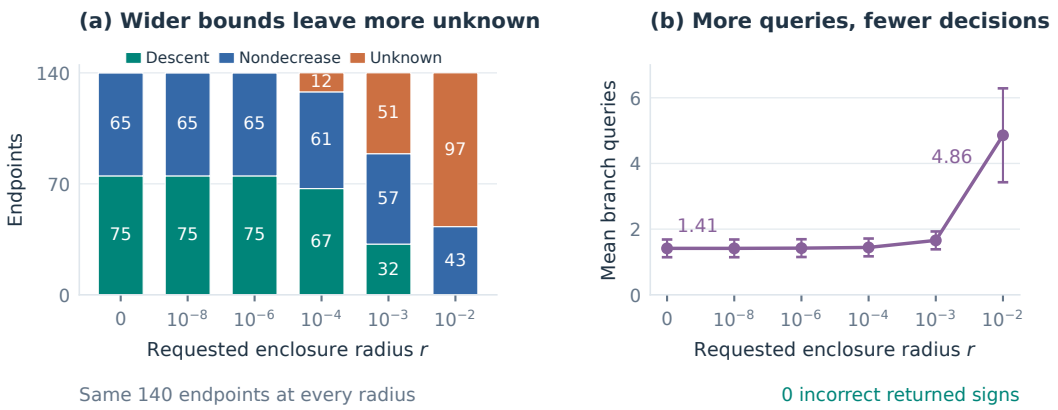


Figure 8: Numerical width changes resolution at fixed endpoints. Left: exact counts of descent, nondecrease, and unknown on the same 140 endpoints, with zero required decrease. Right: mean branch queries and 95% intervals over twenty seeds, each averaged across seven update scales. Radii are discrete experimental settings; zero retains the roundoff floor. All returned signs agree with enumeration. These endpoint records contain no fallback or timing measurements.

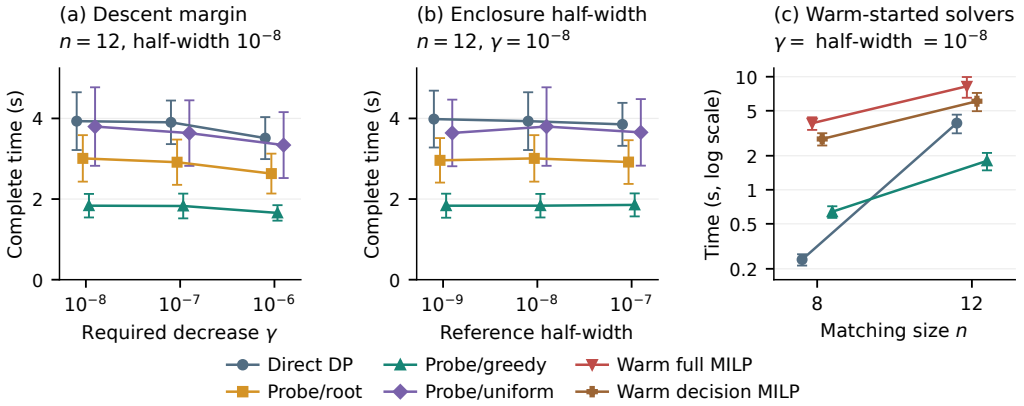


Figure 9: Complete learning cost under different acceptance margins, reference padding, and solver choices. Each point averages the two tail masses within seed, then ten seeds with 95% Student intervals. Warm-started full and decision MILP reuse the model and receive a feasible primal start.

positions and follows Section C.3. A proposal is accepted when $U_1 < L_0 - \gamma$; positions after halting carry zero further work. Changing r pads native-reference enclosures while keeping the branch/assignment arithmetic guards and quantization fixed.

Time, work, and uncertainty. Complete time includes initialization, gradients, branch evaluations, assignment completion, endpoint solving, and baseline refreshes. All global solver calls include initialization and refreshes. The following tables give means over ten independent seeds; time entries include 95% Student intervals. For the pooled comparisons in Figure 9, we first average the two tail masses within each seed. A fallback fraction divides DP fallbacks by proposed endpoints within each trajectory. Query quantiles are computed over active iterations, including their rejected proposals, and then averaged over seeds; they exclude the initial cache evaluation.

Resolution and downstream work. Table 9 connects the fraction requiring DP fallback to the remaining unknown decisions, total solver work, and complete time. At fixed $\gamma = 10^{-8}$, greedy's fallback rate stays close to 11.1% over $r = 10^{-9}, 10^{-8}, 10^{-7}$. The paired increase from 10^{-8} to 10^{-7} is 0.032 ± 0.071 percentage points, with 0.350 ± 0.478 additional DP calls and 0.020 ± 0.034

Tail mass	γ	Half-width	Direct DP	Probe/root	Probe/greedy	Probe/uniform
0.75	10^{-8}	10^{-9}	5.638 ± 1.003	4.013 ± 0.663	2.219 ± 0.355	4.874 ± 1.124
0.75	10^{-8}	10^{-8}	5.593 ± 1.011	4.121 ± 0.728	2.236 ± 0.363	5.256 ± 1.436
0.75	10^{-8}	10^{-7}	5.602 ± 0.887	3.961 ± 0.723	2.218 ± 0.335	4.835 ± 1.165
0.75	10^{-7}	10^{-8}	5.515 ± 0.979	3.936 ± 0.729	2.229 ± 0.380	4.904 ± 1.188
0.75	10^{-6}	10^{-8}	5.089 ± 1.176	3.714 ± 0.989	2.030 ± 0.377	4.605 ± 1.483
0.75	10^{-6}	10^{-7}	4.894 ± 1.111	3.592 ± 0.768	2.060 ± 0.360	4.355 ± 1.169
0.9	10^{-8}	10^{-9}	2.330 ± 0.681	1.907 ± 0.627	1.451 ± 0.357	2.406 ± 0.779
0.9	10^{-8}	10^{-8}	2.272 ± 0.662	1.896 ± 0.623	1.435 ± 0.339	2.340 ± 0.766
0.9	10^{-8}	10^{-7}	2.104 ± 0.535	1.874 ± 0.595	1.492 ± 0.362	2.473 ± 0.817
0.9	10^{-7}	10^{-8}	2.292 ± 0.662	1.893 ± 0.638	1.427 ± 0.337	2.368 ± 0.781
0.9	10^{-6}	10^{-8}	1.936 ± 0.475	1.549 ± 0.427	1.283 ± 0.272	2.073 ± 0.530
0.9	10^{-6}	10^{-7}	2.029 ± 0.515	1.612 ± 0.453	1.320 ± 0.277	2.065 ± 0.485

Table 7: Complete time in seconds for all twelve tolerance/tail-mass configurations. Each cell is a ten-seed mean and 95% interval.

β	γ	Half-width	DP calls: D/R/G/U	Fallback: R/G/U (%)	G queries: 50/90/99
0.75	10^{-8}	10^{-9}	371.6/250.2/71.0/204.1	57.6/9.7/44.4	6.4/15.6/21.0
0.75	10^{-8}	10^{-8}	371.6/250.2/71.0/204.1	57.6/9.7/44.4	6.4/15.6/21.0
0.75	10^{-8}	10^{-7}	367.1/250.1/71.6/203.0	57.1/9.7/43.8	6.4/15.7/20.9
0.75	10^{-7}	10^{-8}	364.4/250.4/71.6/203.8	57.2/9.7/44.0	6.4/15.7/20.9
0.75	10^{-6}	10^{-8}	338.4/225.2/68.2/189.4	56.8/10.7/45.0	6.0/15.4/20.5
0.75	10^{-6}	10^{-7}	321.6/222.0/67.1/186.9	57.4/10.7/45.5	6.0/15.3/20.9
0.9	10^{-8}	10^{-9}	256.6/190.8/59.9/129.6	61.2/12.5/38.2	4.8/11.6/17.9
0.9	10^{-8}	10^{-8}	256.6/190.8/59.7/129.4	61.2/12.5/38.2	4.8/11.6/17.8
0.9	10^{-8}	10^{-7}	226.5/188.3/59.8/129.5	60.7/12.5/38.2	4.8/11.6/17.8
0.9	10^{-7}	10^{-8}	256.6/191.3/59.7/129.4	61.3/12.5/38.2	4.8/11.6/17.8
0.9	10^{-6}	10^{-8}	208.5/148.2/54.5/109.8	58.1/13.5/40.1	4.0/10.4/18.7
0.9	10^{-6}	10^{-7}	212.4/155.0/54.6/107.8	59.1/13.6/39.9	4.0/10.4/18.8

Table 8: Mean complete DP calls, fallback percentages, and greedy branch-query quantiles per active iteration. D, R, G, and U denote direct DP, probe/root, probe/greedy, and probe/uniform; the ordered entries follow the header. Quantiles are the 50th, 90th, and 99th percentiles.

seconds per trajectory. These intervals include zero; the sweep does not establish a runtime penalty over this padding range. Eighteen of the twenty cross-radius greedy parameter paths coincide.

At fixed $r = 10^{-8}$, raising γ from 10^{-7} to 10^{-6} increases the fallback fraction by 1.016 ± 0.862 percentage points, yet reduces DP calls by 4.300 ± 4.158 and time by 0.172 ± 0.156 seconds. Only six of twenty cross-margin paths coincide. The stricter acceptance condition changes the trajectories and their lengths, so a larger fallback fraction need not imply more total work. These estimates average the two tail masses within each seed before forming paired ten-seed intervals.

These measurements resolve the total effect of fallback and current-risk refinement. They do not identify how much of that effect comes from the first branch query, the root bound, or later tree search, nor distinguish numerical overlap from an exhausted search budget as the reason for fallback.

Tolerance effects and matched paths. Across the six settings, the ratio of direct-DP to greedy mean time is 2.05–2.17. All six paired greedy time-saving intervals exclude zero; the six corresponding uniform intervals include zero. At $(\gamma, r) = (10^{-8}, 10^{-8})$ and $(10^{-8}, 10^{-9})$, every policy has the same saved parameter path as direct DP. Greedy matches 18/20 paths at $(10^{-7}, 10^{-8})$, 19/20 at $(10^{-6}, 10^{-8})$, 14/20 at $(10^{-8}, 10^{-7})$, and 15/20 at $(10^{-6}, 10^{-7})$. At the last, jointly conservative setting, the complete paired saving is 1.771 ± 0.344 seconds; the saving through each pair's common parameter prefix, including initialization, is 1.560 ± 0.338 seconds. We evaluate final risk with the native DP at the saved parameters. Across all sensitivity comparisons, the largest absolute individual terminal-risk difference from direct DP is 3.27×10^{-4}.

γ	DP radius r	Fallback (%)	Unknown (%)	DP calls	Branch calls	Time (s)
10^{-8}	10^{-9}	11.09	0.00	65.45	560.2	1.835 ± 0.298
10^{-8}	10^{-8}	11.07	0.00	65.35	560.0	1.836 ± 0.292
10^{-8}	10^{-7}	11.10	0.05	65.70	561.5	1.855 ± 0.285
10^{-7}	10^{-8}	11.07	0.02	65.65	561.5	1.828 ± 0.307
10^{-6}	10^{-8}	12.09	0.50	61.35	490.1	1.656 ± 0.192
10^{-6}	10^{-7}	12.15	0.56	60.85	484.4	1.690 ± 0.180

Table 9: Greedy search with DP fallback at 12×12. Rates are computed per trajectory; each mean averages the two tail masses within each seed and then ten seeds. Time intervals are 95% Student intervals. DP and branch counts include initialization and refreshes; unknown denotes the final proposal status after fallback and any baseline refresh. The varied radius pads DP/reference enclosures; tree arithmetic remains fixed.

Warm-started MILP references. Both MILP policies reuse their model and receive the previous feasible retained set with tail weights recomputed at the proposed endpoint. A warm start supplies the preceding feasible retained set as an initial solution. Full MILP closes the objective bounds; decision MILP stops when either the best feasible value or the global upper bound settles the prescribed comparison. An unresolved comparison can refresh a loose current baseline. Tables 10 and 11 retain each size and tail mass. All four policies follow identical parameter paths in all forty data/size/tail-mass configurations. At $n = 12$, greedy saves 2.086 ± 0.432 seconds against direct DP, while warm decision and full MILP cost 2.184 ± 0.459 and 4.340 ± 1.022 seconds more than DP, respectively. At $n = 8$, direct DP remains fastest. Solver-node counts characterize individual endpoint solves; total time and call counts measure the complete training cost.

n	β	Direct DP	Warm full MILP	Warm decision MILP	Probe/greedy
8	0.75	0.308 ± 0.076	4.382 ± 1.151	3.203 ± 0.720	0.766 ± 0.192
8	0.9	0.175 ± 0.046	3.379 ± 1.269	2.426 ± 0.754	0.513 ± 0.157
12	0.75	5.554 ± 1.043	9.384 ± 1.734	6.678 ± 1.123	2.206 ± 0.380
12	0.9	2.222 ± 0.669	7.071 ± 2.242	5.465 ± 1.444	1.398 ± 0.356

Table 10: Complete learning time in seconds with four exact-solver choices. Entries are ten-seed means and 95% intervals.

n	β	DP calls: D/G	MILP calls: F/S	Accepted	Active	Proposals
8	0.75	397.9/74.0	397.9/413.5	87.0	87.4	396.9
8	0.9	286.4/60.0	286.4/300.9	75.1	76.0	285.4
12	0.75	371.6/71.0	371.6/390.7	86.4	87.1	370.6
12	0.9	256.6/59.7	256.6/274.1	79.3	80.3	255.6

Table 11: Complete solver calls and mean learning counts. D/G denotes direct DP/greedy; F/S denotes warm full/decision MILP. Accepted, active, and proposal counts are identical across the four policies in each row.

Scale dependence by tail mass. Table 12 separates the two tail masses in the earlier twenty-seed experiment of Section D.5. Direct DP leads at $n = 6, 8$. At $n = 10$, probe/root leads for $\beta = 0.75$ while direct DP leads for $\beta = 0.9$; greedy leads for both tail masses at $n = 12$. Thus direct DP is preferable at the two smaller scales, while the choice at the intermediate scale depends on tail mass.

n	β	Direct DP	Probe/root	Probe/greedy	Probe/uniform
6	0.75	0.072 ± 0.011 (257)	0.120 ± 0.019 (178)	0.301 ± 0.048 (61)	0.398 ± 0.087 (62)
6	0.9	0.063 ± 0.009 (232)	0.112 ± 0.022 (158)	0.268 ± 0.061 (56)	0.334 ± 0.090 (58)
8	0.75	0.233 ± 0.031 (352)	0.273 ± 0.038 (233)	0.600 ± 0.090 (68)	1.027 ± 0.207 (110)
8	0.9	0.153 ± 0.020 (277)	0.211 ± 0.035 (196)	0.501 ± 0.082 (63)	0.741 ± 0.166 (87)
10	0.75	0.914 ± 0.157 (324)	0.751 ± 0.118 (213)	0.851 ± 0.124 (64)	1.462 ± 0.276 (134)
10	0.9	0.312 ± 0.035 (226)	0.326 ± 0.038 (154)	0.627 ± 0.082 (56)	0.927 ± 0.147 (95)
12	0.75	5.604 ± 0.587 (393)	3.959 ± 0.390 (262)	2.186 ± 0.176 (75)	4.779 ± 0.582 (212)
12	0.9	2.445 ± 0.368 (302)	1.911 ± 0.276 (211)	1.459 ± 0.166 (65)	2.522 ± 0.368 (149)

Table 12: Original matching study, seeds 260–279: complete time in seconds (mean $\pm$ 95% interval), followed by mean complete DP calls in parentheses. All scales and both tail masses are retained.

D.5 Coupled row and column group constraints

This comparison couples row and column choices, so a fixed-threshold bound uses assignment whereas an exact endpoint solve also allocates tail weights. The generator supplies n^2 groups of twelve observations, retaining one per row and column. Grid sizes $n = 6, 8, 10, 12$, tail masses $0.75, 0.9$, seeds 260–279, and four policies give 640 trajectories. Direct DP, probe/root, probe/greedy, and probe/uniform use the same native reference and Section C.3. Assignment completion and relaxed LPs are charged; a separate full MILP supplies reference intervals. Scale estimates average tail masses within seed before forming twenty-seed intervals.

Direct DP leads at $n = 6, 8$, probe/root at 10, and greedy at 12. At $n = 12$, their respective times are 4.025 ± 0.365, 2.935 ± 0.261, 1.823 ± 0.132, and 3.651 ± 0.378 seconds. Greedy and uniform save 2.202 ± 0.249 and 0.374 ± 0.241 seconds against DP; greedy additionally saves 1.112 ± 0.138 against probe/root. Greedy wins all forty size-twelve configurations; uniform wins 26. At size twelve, DP calls fall from 347.2 to 69.8 for greedy and 180.2 for uniform, while their assignment counts are 77,434.4 and 128,527.3. Thus discovery order affects both saved exact calls and completion overhead.

Of 480 nonreference/DP path pairs, 470 are identical. Ten comparisons differ in four configurations; at size twelve only seed 271, $\beta = 0.75$, differs, starting at iteration 96. Size-twelve terminal risks differ by at most 2.97×10^{-6}. Shared-prefix paired savings, including initialization, remain 2.197 ± 0.250 seconds for greedy and 0.374 ± 0.241 for uniform. Across all sizes the largest terminal difference is 6.67×10^{-5}, with mean paired difference $-5.76 \times 10^{-7} \pm 8.89 \times 10^{-7}$ for either search policy.

Reference values use enumeration at $n \leq 8$ and a separate reverse-row full-tail long-double DP at $n = 10, 12$. All 49,236 accepted changes decrease the objective: the largest enclosed change upper bound is -2.65×10^{-8}, with maximum reference allowance 5.19×10^{-12}. Section C.10 gives the pathwise error bound for the implementation's complementary-count formulation.

n	Policy	Time (s)	DP calls	Branch solves	Nodes	Assignments
6	Exact DP	0.067	244.3	0.0	0.0	0.0
6	Probe + DP	0.116	168.2	386.9	0.0	999.4
6	Probe + greedy + DP	0.284	58.5	538.0	1899.1	21892.1
6	Probe + uniform + DP	0.366	59.8	550.7	2822.1	30046.0
8	Exact DP	0.193	314.4	0.0	0.0	0.0
8	Probe + DP	0.242	214.8	497.5	0.0	1273.3
8	Probe + greedy + DP	0.551	65.9	631.0	3764.2	43328.3
8	Probe + uniform + DP	0.884	98.8	569.6	7079.1	78787.0
10	Exact DP	0.613	275.2	0.0	0.0	0.0
10	Probe + DP	0.539	183.5	429.3	0.0	1100.1
10	Probe + greedy + DP	0.739	60.0	521.1	4364.2	48665.9
10	Probe + uniform + DP	1.195	114.6	458.9	7510.3	85004.6
12	Exact DP	4.025	347.2	0.0	0.0	0.0
12	Probe + DP	2.935	236.4	550.2	0.0	1413.0
12	Probe + greedy + DP	1.823	69.8	611.8	6781.4	77434.4
12	Probe + uniform + DP	3.651	180.2	556.5	11055.5	128527.3

Table 13: Complete policy costs across all matching scales. Entries are means per trajectory over forty data/tail-mass configurations at each scale. Global DP calls include initialization, endpoint fallback, and baseline refinement. Assignment completion and explicit branch work are charged separately.

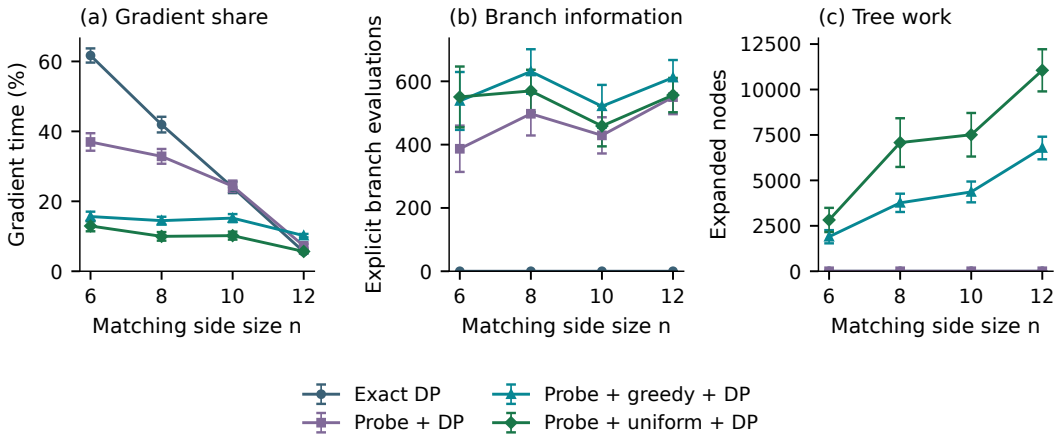


Figure 10: Work components in matching-constrained learning. The first panel shows the fraction of complete policy time spent computing relaxed gradients. Branch evaluations include the feasible probe, cache updates, and tree leaves. Node counts measure the budgeted tree work. Points and intervals aggregate within seed before forming twenty-seed means and 95% intervals.

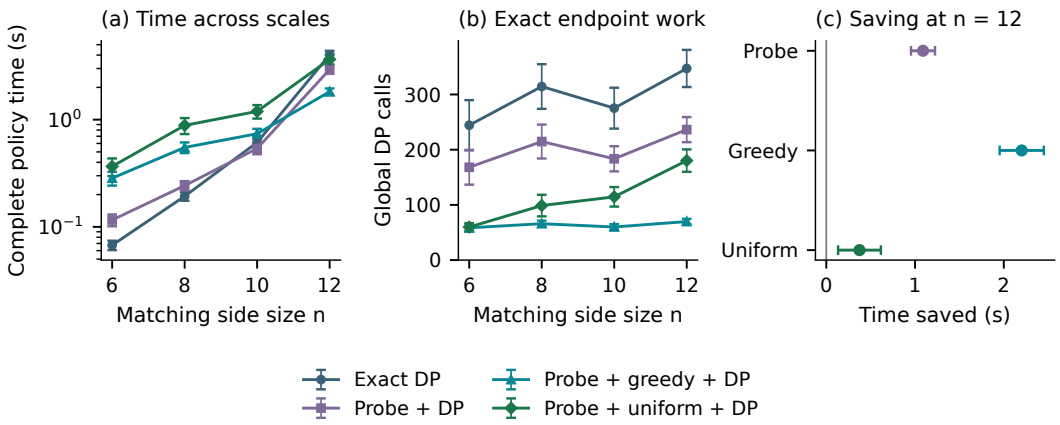


Figure 11: Complete learning under row/column retention. Left: total policy time. Middle: global DP calls. Right: paired time saved against direct DP at $n = 12$. Timers include initialization, gradients, branch and assignment work, and refinements. Points show twenty-seed means and 95% intervals, averaging the two tail masses within seed.

D.6 One 64-group learning family with three exact solvers

All three studies below retain four of 64 groups of 24 observations, at tail masses $0.75, 0.9$, freely or with balanced quotas. Their 635,376 or 246,016 retained sets share the generator and controller of Section C.3. Each uses twenty fresh seeds and four configurations per seed. The successive comparisons isolate full fallback, decision fallback and probes, then a specialized polynomial reference. Independent streaming enumeration checks their saved value enclosures and accepted changes.

Full-fallback comparison. This comparison asks whether shared bounds save time when unresolved updates require a full MILP solve. Seeds 200–219 give 640 trajectories. Controls accept every proposal, check every fifth position, or require hull decrease. Periodic checking accepts unchecked proposals and continues after a failed scheduled check. The five discrete policies use full MILP, decision MILP, or cached root bounds, greedy search, or uniform search with full fallback. The cached variants start from stored thresholds and the relaxed threshold, without the separate feasible-branch probe used in the next study.

The five discrete policies accept 5,746–5,763 updates per policy with zero increases. Mean terminal risks agree within 5.0×10^{-7}; the largest matched difference from full MILP is 3.10×10^{-5}. Greedy's paired saving is 2.208 ± 0.600 seconds against full MILP and 2.086 ± 0.495 against cached screening; it wins 75/80 and 71/80 configurations. Uniform saves 0.662 ± 0.503 and 0.540 ± 0.423. Decision MILP has the lowest aggregate mean time, 3.608 seconds; greedy and uniform exceed it by 0.189 ± 1.389 and 1.735 ± 1.283 seconds.

Always-accept and periodic controls finish at exact risks 1.8783 and 1.8779 versus full checking's 1.8889, admitting 2,724 and 2,444 increases. Hull search reaches 1.8871 with 1,072 increases. These are final objectives at 100 positions with stopped paths carried forward; Section D.1 supplies the comparison at a common stopping position.

Decision fallback and feasible probes. We next separate the benefit of a first branch probe from stopping the fallback solver as soon as its bounds decide the comparison. Seeds 220–239 give 720 trajectories. The nine policies are full/decision MILP, cached root with decision fallback, greedy with full/decision fallback, uniform with decision fallback, and probe/root, probe/greedy, or probe/uniform with decision fallback. The probe is charged before discovery.

Probe/greedy takes 2.138 seconds versus 3.092 for decision MILP and 5.275 for full MILP. Its paired saving against decision MILP is 0.955 ± 0.232 seconds, with wins in 71/80 configurations. It saves 0.789 ± 0.215 against cached screening and 0.641 ± 0.171 against probe/root. Greedy without the probe takes 2.172 seconds; the probe adds a smaller saving of 0.034 ± 0.017. Switching greedy's

Policy	Final risk	Increases	Time (s)	MILP calls
Relaxed	1.8783	2,724	0.236	0.0
Periodic (5)	1.8779	2,444	1.518	50.3
Hull line search	1.8871	1,072	0.760	0.0
Full MILP	1.8889	0	6.005	284.5
Decision MILP	1.8889	0	3.608	300.3
Screen + MILP	1.8889	0	5.883	271.9
Greedy + MILP	1.8889	0	3.797	105.7
Random + MILP	1.8889	0	5.344	201.0

Table 14: Complete learning with 64 groups, 80 trajectories per policy. Final risk, runtime, and MILP calls are means per trajectory. Increasing accepted steps are totals at tolerance 10^{-8}. Timings include initialization, relaxed gradients, baseline refinements, and endpoint checks.

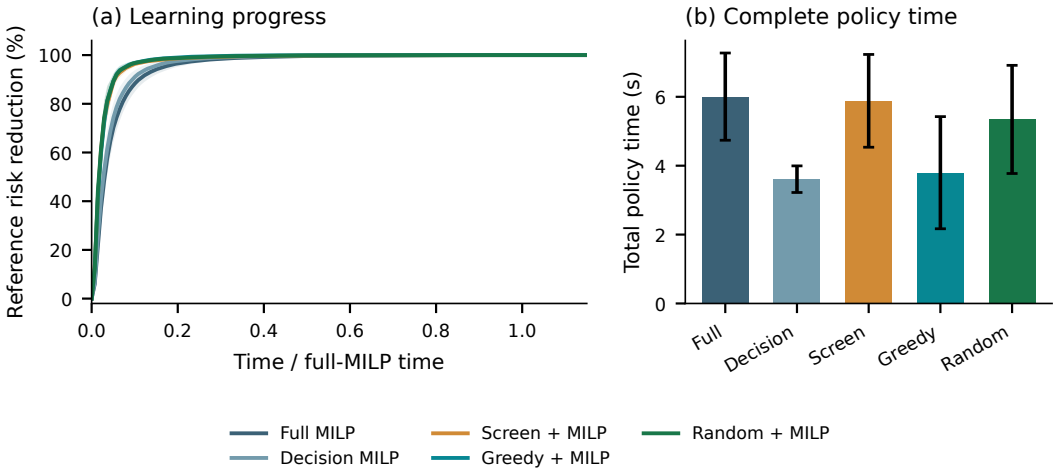


Figure 12: The full-fallback learning comparison with 64 groups. Left: achieved risk reduction as a percentage of the matched full-MILP trajectory's reduction, against time normalized by that trajectory's total time. Right: complete policy runtime, including initialization and refinements. Bands and bars are 95% intervals over twenty seed means. Stopped trajectories retain their final values with zero further work.

fallback from full to decision MILP reduces mean time from 3.349 to 2.172 seconds. Thus directed exploration and solver stopping supply distinct gains.

Probe/greedy settles 14,292 of 20,719 comparisons before fallback, versus 2,159 for cached screening. Its mean work is 668.3 branch solves, 6,642.5 expansions, and 104.1 MILP calls, versus decision checking's 273.3 MILP calls. All nine methods have identical paths and mean terminal risk 1.893954.

Specialized dynamic programming. To test whether the gains depend on an expensive fallback solver, replace MILP by the specialized exact DP for disjoint groups. Seeds 240–259 give 320 trajectories across four policies using Section B.8's native implementation. Sorting, allocation, recurrence, and witness reconstruction are timed; compilation precedes timing. A separate full MILP supplies reference intervals.

Direct DP takes 0.235 ± 0.021 seconds per trajectory. Probe/root, greedy, and uniform add 0.061 ± 0.013, 0.208 ± 0.053, and 0.356 ± 0.081 seconds, respectively; they are faster in 4, 3, and 1 of the 80 configurations. Greedy reduces global calls from 255.9 to 104.6, but adds 668.4 branch solves and 6,745.2 expansions. The saved exact calls do not offset search overhead when the specialized reference is inexpensive.

All four policies have identical parameters, acceptance masks, and trial counts, with mean terminal risk 1.865914. Witnesses reconstruct tail weights and a value subgradient (Danskin, 1967).

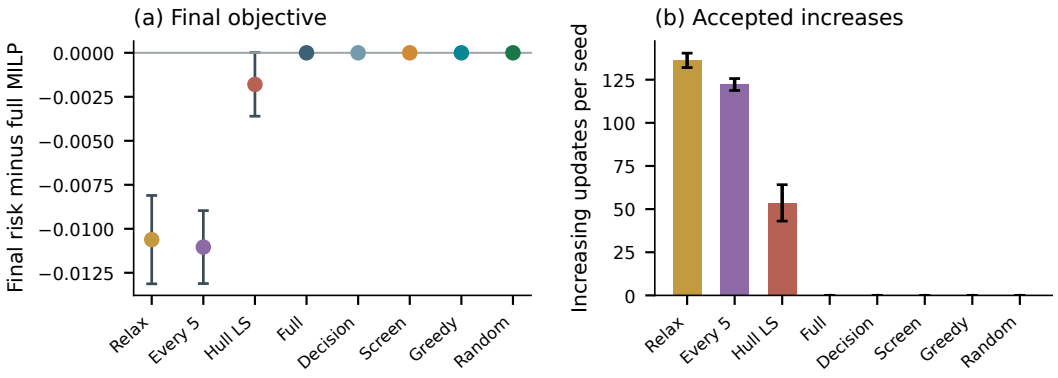


Figure 13: All eight policies in the 64-group learning experiment. Left: paired terminal risk relative to full MILP checking. Right: increasing accepted updates per seed, summed over four configurations. Error bars are 95% intervals over twenty independent seed means. The five discrete checking policies produce zero increasing accepted updates.

Policy	Final risk	Time (s)	Branch solves	Nodes	MILP calls
Full MILP	1.893954	5.275	0.0	0.0	260.0
Decision MILP	1.893954	3.092	0.0	0.0	273.3
Screen + decision	1.893954	2.927	260.0	0.0	250.6
Greedy + full	1.893954	3.349	495.6	7299.1	105.8
Greedy + decision	1.893954	2.172	495.9	7308.1	107.0
Random + decision	1.893954	2.821	364.4	12358.7	181.4
Probe + decision	1.893954	2.779	460.9	0.0	219.5
Probe + greedy	1.893954	2.138	668.3	6642.5	104.1
Probe + random	1.893954	2.587	538.6	9934.6	153.4

Table 15: Nine-policy decision-oriented learning comparison. All entries are means per trajectory across 80 configurations. Every policy accepts 5,497 updates in total, with zero increasing accepted updates and identical matched parameter trajectories. Branch counts include explicit probe, cache updates, lower-bound evaluations, and tree queries; internal evaluations of feasible retained sets are charged to the MILP calls.

Policy	Final risk	Time (s)	Branch solves	Nodes	DP calls
Exact DP	1.865914	0.235	0.0	0.0	255.9
Probe + DP	1.865914	0.296	454.1	0.0	213.5
Probe + greedy + DP	1.865914	0.442	668.4	6745.2	104.6
Probe + random + DP	1.865914	0.590	525.0	10253.1	159.4

Table 16: Complete learning with the native exact-DP reference. All entries are means per trajectory over 80 configurations. Every policy has the same matched parameter path, 5,558 accepted updates in total, and zero increasing accepted updates. DP calls include initialization, refinements, and endpoint work. Explicit branch calls and tree nodes are counted separately.

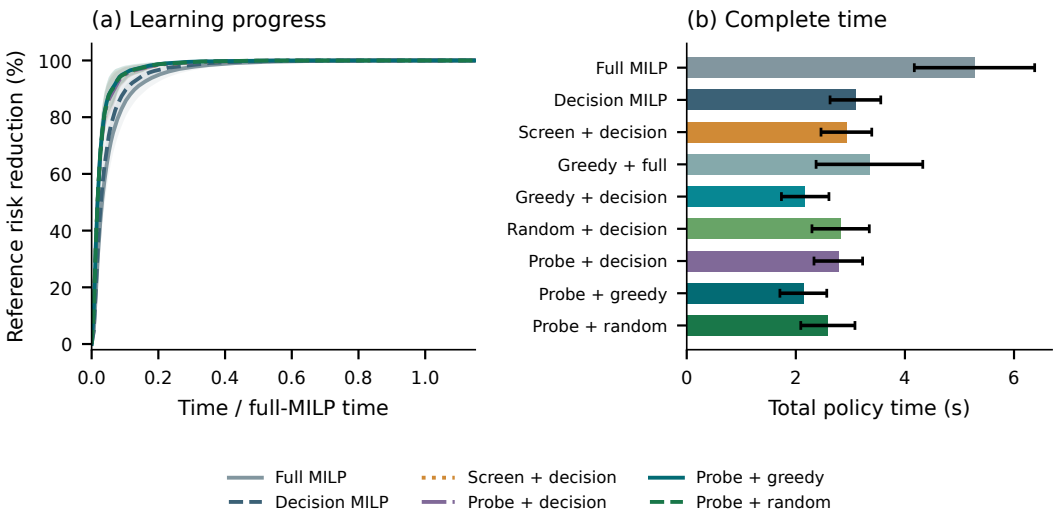


Figure 14: Decision-oriented learning with 64 groups. Left: achieved risk reduction relative to the matched full-MILP trajectory, against its normalized total time. Right: complete policy times for all nine methods. All probe variants use decision fallback. Bands and bars are 95% intervals over 20 seed means; initialization, refinements, and probes are charged. Stopped trajectories retain their final values with zero further work.

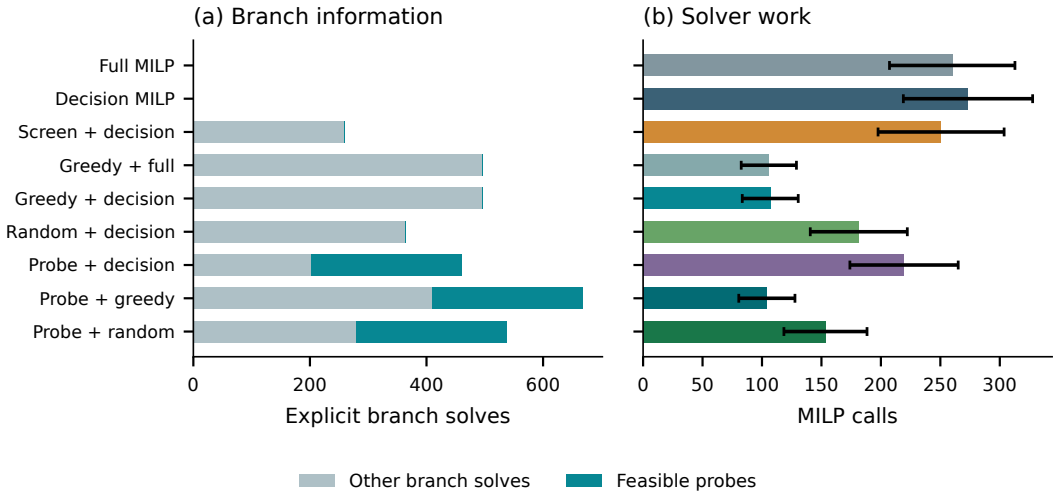


Figure 15: Work in the nine-policy learning comparison. Left: explicit branch solves per trajectory, separating deterministic feasible probes from cache updates, lower-bound evaluations, and tree queries. Right: MILP calls including initialization and baseline refinements, with 95% intervals over data-seed means. Evaluations of feasible retained sets inside a MILP solve are included in its time and MILP count. All probe variants use decision fallback.

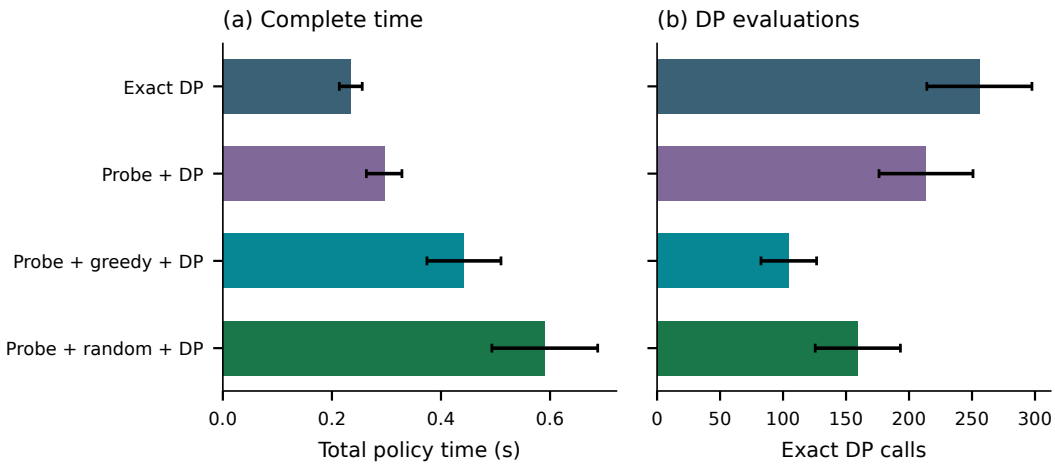


Figure 16: Complete learning for disjoint group quotas with native exact DP. Left: total policy time. Right: global DP calls, including initialization and refinements. Branch queries, tree work, and gradient solves enter the timer. All four policies produce identical matched parameter trajectories. Bars show means and 95% intervals over twenty data seeds.

D.7 Training acceptance policies

Twelve configurations combine deletion budgets $4, 8, 10$, tail masses $0.75, 0.9$, and both retention constraints. Twenty datasets and six policies give 1,440 trajectories of at most 100 positions, proposing $-\eta g_H$ with initial $\eta = 1.5/L_{\text{sample}}$ and twelve-step halving. The relaxed policy accepts the initial proposal; periodic checking enumerates every fifth iteration; hull line search requires relaxed decrease above 10^{-8}; exact checking enumerates every endpoint against its cached exact baseline. A failed backtracking sequence halts hull and exact checking.

Greedy and uniform sharing initialize the exact baseline and then carry accepted enclosures, targeting decrease 10^{-8}. An accepted proposal with no exact endpoint query triggers one known feasible-branch evaluation to establish a lower bound at the next iterate. After twelve failures with a loose baseline, sharing refreshes it by enumeration and retries the same sequence once; failure with an exact baseline halts the trajectory. Stopped positions retain risk with zero further work. The uniform stream is fixed by dataset seed, configuration, and policy seed 3107. Runtime and branch-query counts include initialization, evaluations establishing lower bounds, queries, and refreshes; reference enumeration follows timing.

Exact, greedy, and uniform checking use means of 28.45, 28.27, and 27.92 active positions. Paired terminal-risk differences from exact checking are -0.000725 ± 0.001899 for greedy and -0.000591 ± 0.001916 for uniform (95% intervals).

Policy	Final risk	Increasing steps	Branch evaluations	Time (ms)
Relaxed	1.5357	10,284	0	9.22
Periodic (5)	1.5359	9,309	14,886	12.93
Hull line search	1.5156	1,721	0	15.43
Exact check	1.5204	0	30,449	10.31
Greedy share	1.5197	0	4,790	32.05
Random share	1.5198	0	5,166	40.84

Table 17: Training over 20 evaluation seeds and 12 configurations. Final risk, branch evaluations, and runtime are means per trajectory. Increasing steps are totals across the 240 trajectories of each policy; stopped trajectories incur no further work.

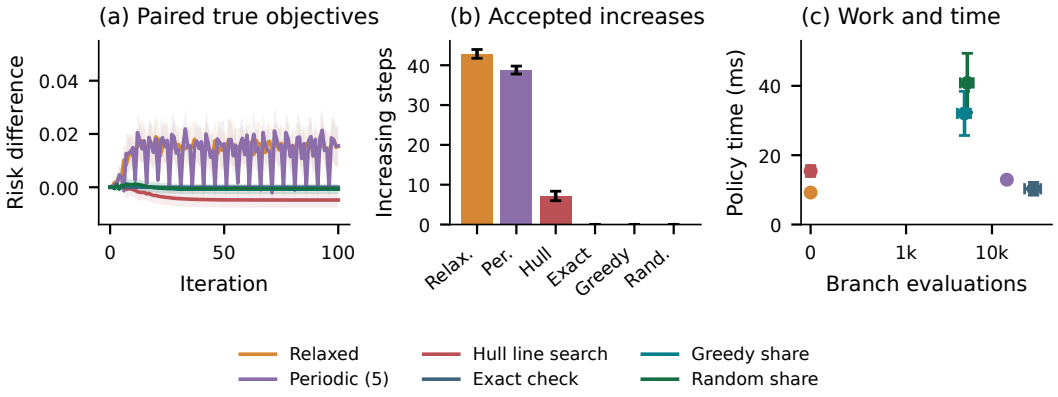


Figure 17: Six acceptance policies in the small-family learning experiment. Left: true risk minus the matched exact-check trajectory. Middle: increasing accepted updates per trajectory. Right: charged branch work and policy runtime, including initialization and refreshes. Curves are paired by dataset and risk configuration; bands summarize data-seed uncertainty. Reference enumeration is performed after timing.

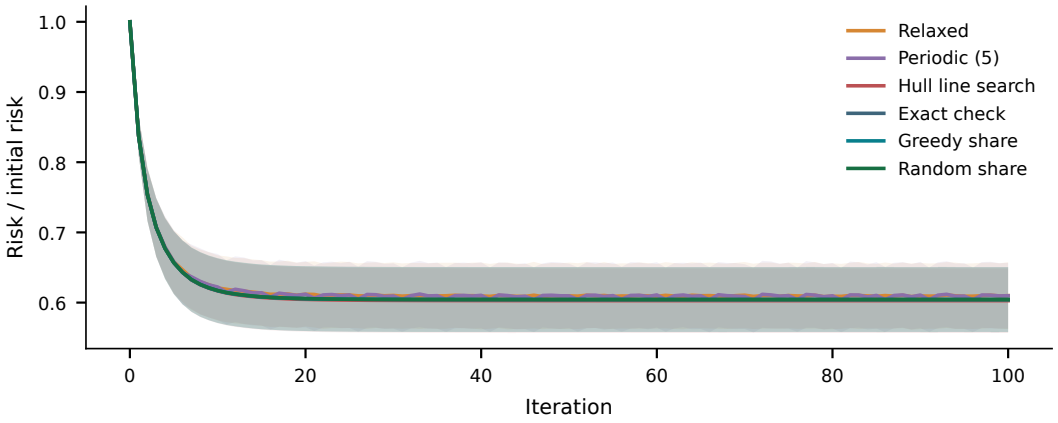


Figure 18: True training risks relative to each trajectory's initial risk in the small-family experiment. Bands are confidence intervals over independent seed means. The paired comparison in Figure 17 removes shared dataset variation to show differences among acceptance policies.

D.8 Larger families with MILP references

For $G \in \{12, 16, 20, 24, 28, 32\}$, retain $k = G/2$, use tail masses $0.35, 0.65$ and both constraints, and generate fresh datasets with seeds 100–109. After 30 relaxed path steps, propose scales $0.15, 1.5$. All 480 endpoints are checked by single-threshold and shared-threshold trees and MILP; both trees finish within 20,000 node expansions. Current and endpoint MILPs reach optimal status with one thread, feasibility/optimality tolerances 10^{-9}, relative/absolute gaps 10^{-10}, and a 120-second limit. Baseline time is recorded separately.

At $G = 32$, shared-tree endpoint time has median 0.0557, 90th percentile 1.2493, and maximum 37.4997 milliseconds; it beats paired MILP on 77/80 endpoints. Baseline solves average 3.1111 milliseconds. The 38 cases requiring tree expansion have median 0.7540 and mean 2.9850 milliseconds, with at most 1,152 expansions. Result tables retain results with and without tree expansion at every size.

To separate input size from the number of feasible branches, the observation/dimension study varies group count G, observations per group s_g, and input dimension d. It uses $(G, s_g, d) = (24, 4, 6), (24, 16, 16), (24, 32, 32), (32, 32, 32)$. Ten seeds, both constraints, tail masses $0.65, 0.9$, and both scales give 320 endpoints. The same generator extends the concentrated profile and alternating polarity to each group size.

Groups	Max. branches	Root decisions (%)	Leaves	Nodes	Time (ms)
12	924	53.8	0.51	3.27	0.238
16	12,870	51.2	0.60	4.90	0.249
20	184,756	50.0	0.56	5.59	0.268
24	2,704,156	55.0	0.51	6.49	0.284
28	40,116,600	56.2	0.57	9.09	0.435
32	601,080,390	52.5	0.60	40.62	1.440

Table 18: Shared-threshold tree decisions. Maximum branch count is the free-cardinality family; the balanced family is smaller. Counts and time are means over each group-count setting. Root decisions explicitly identify proposals resolved before a node expansion.

D.9 Larger groups and branch-access cost

The two settings $(G, s_g, k, d) = (32, 48, 2, 32)$ and $(64, 24, 4, 32)$ each have 1,536 observations and 96 retained per branch. Ten seeds, both constraints, tail masses $0.75, 0.9$, snapshots 80/240, and scales $0.05, 0.5$ give 160 endpoints per setting. Full current and endpoint MILPs reach optimal status

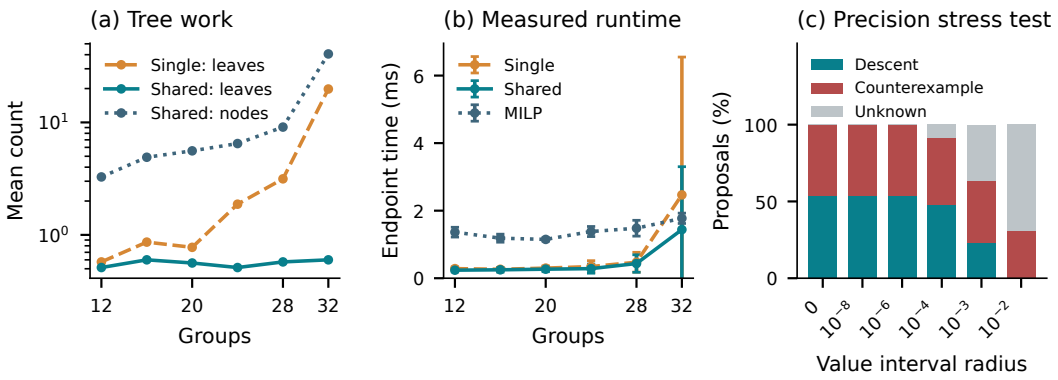


Figure 19: Group-count and precision studies. Left: leaf queries and expanded nodes from 12 to 32 groups. Middle: shared and single-threshold tree runtime compared with MILP. Right: decisions as the enclosure radius increases. Curves and uncertainty aggregate independent data seeds.

with integer-feasibility tolerance 10^{-9}; reevaluation of feasible branches and solver bounds receive outward allowance $r = 10^{-8}$. Streaming enumeration provides independent reference values for these intervals.

At 32 groups, the free and balanced families have 496 and 256 branches. We compare full sorting, partial-selection enumeration, greedy/uniform sharing, and implicit trees. Partial selection finds the fractional tail boundary without fully sorting the losses; sharing uses the same branch-evaluation routine. At 64 groups, the respective family sizes are 635,376 and 246,016, and the comparison uses trees, full MILP, and decision MILP. Across both settings, 204 endpoints decrease and 116 supply counterexamples; 137/160 and 109/160 require tree expansion.

Each endpoint is replayed seven times with shuffled method order; randomized discovery retains its fixed query stream. A MILP model is restored to its current baseline before endpoint timing. The timer includes losses, bounds, scans, queries, and solving. We take the median per endpoint, then average configurations within each of ten seeds for uncertainty estimates.

At 32 groups, uniform sharing uses 1.93 branch queries versus 376 for enumeration. Uniform, greedy, partition enumeration, and sorting average 0.1919, 0.1590, 0.2077, and 0.4619 milliseconds. Greedy uses 1.73 queries. Paired partition-minus-sharing savings are 0.0158 ± 0.0098 milliseconds for uniform and 0.0487 ± 0.0050 for greedy; sharing is faster on 93/160 and 126/160 endpoints. Among 137 nonroot endpoints, uniform and partition means are 0.2059 and 0.2072 milliseconds.

Decision MILP stops when its best feasible objective value reaches $U_0 + 2r - R(y)$ or its upper objective bound falls below $L_0 - 2r - R(y)$. These feasible solutions are reevaluated as branch risks and returned bounds receive allowance r. Across seven replays, decision MILP stops at its target in 700/1,120 calls and reaches optimal status in 420; every returned sign agrees with the reference.

At 64 groups, the tree averages 1.34 leaf queries and 47.45 expansions. Tree, decision MILP, and full MILP take 3.5122, 16.0200, and 21.0449 milliseconds. The paired decision-minus-tree saving is 12.5078 ± 1.4047 milliseconds ($4.56\times$ ratio of means); among nonroot endpoints it is 13.8430 ± 1.8942 ($3.96\times$), with method times 4.6838, 18.5267, and 24.5131 milliseconds. The tree beats decision MILP on 157/160 endpoint medians. These contrasts retain the cost of branch access alongside its query count.

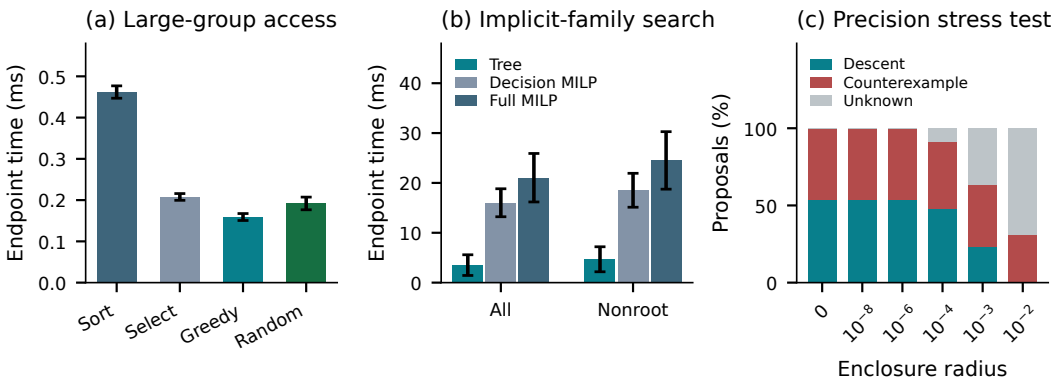


Figure 20: Computation and accuracy. Left: 32 groups of 48 observations, retaining two; enumeration uses full sorting or partial selection. Middle: 64 groups of 24 observations, retaining four; decision MILP stops once its bounds establish the sign. Times use seven-repetition endpoint medians, then means and confidence intervals over ten data seeds. Right: decisions as branch and baseline enclosure radii increase.

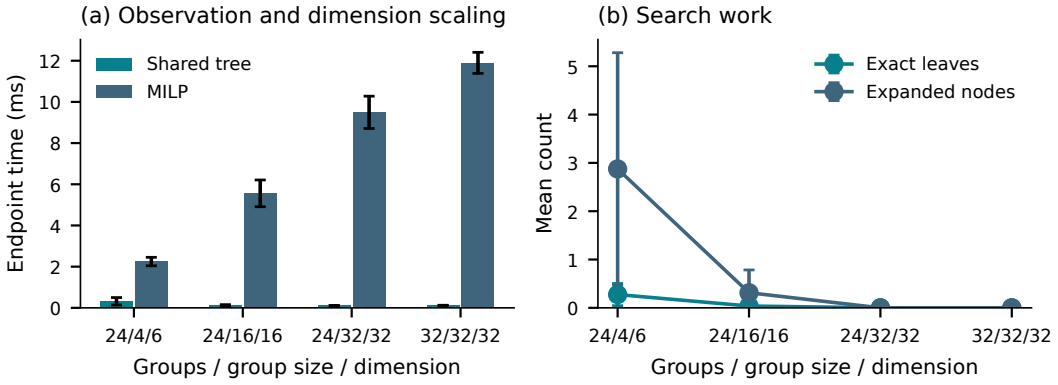


Figure 21: Scaling observations and parameter dimension within the same generator. Each setting has 80 endpoints from ten datasets. Left: complete endpoint time compared with MILP. Right: shared-tree search work. Root decisions account for 81.25%, 97.5%, 100%, and 100% of the four settings; the two 32-dimensional settings measure screening throughput.

D.10 Implicit and flat query discovery

All 320 branch-cost endpoints receive eight implicit uniform runs and one greedy run; the 160 smaller-family endpoints also receive eight matched flat-uniform runs. Node bounds reuse group scores and evaluate only newly added thresholds. Candidate and cover counts are computed after the run. These counts describe the diagnostic endpoints, not the public-data training trajectories. Figure 22 averages policy seeds within endpoint and then data seeds, separating branch queries from node expansions. All decisions and enclosures agree with the streaming reference.

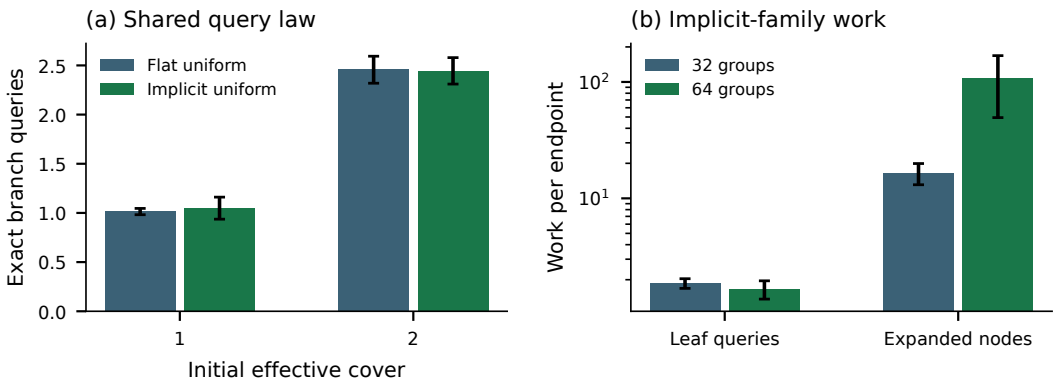


Figure 22: Implicit discovery on the 320 saved branch-cost endpoints. Left: exact branch queries for flat and implicit uniform selection, grouped by the initial effective cover at 32 groups. Policy seeds are averaged within each endpoint before averaging data seeds. Right: leaf queries and expanded nodes for implicit uniform search at 32 and 64 groups. Bars give 95% intervals over contributing data seeds; the right axis is logarithmic.

E SUPPLEMENTARY ANALYSES

These analyses examine threshold sharing, numerical precision, and query budgets on the matched regression proposals.

E.1 HOW NUMERICAL BOUNDS DETERMINE ACCEPTANCE

Response error, residual, and the strict comparison. To accept safely with an approximate threshold, account separately for evaluation error and threshold suboptimality. Let $\widehat{F}$ evaluate a returned threshold with absolute error at most e_F, and let ζ certify $F_B(y,t) - \min_u F_B(y,u) \le \zeta$. Outward rounding gives $[\widehat{F} - e_F - \zeta, \widehat{F} + e_F]$ for that branch's minimum. A shared response uses $\widehat{F} + e_F$, since its threshold is feasible. The CVaR quantile routine directly minimizes the stored-loss problem; the loss and regularizer bounds in Sections C.8 and D.3 transfer its value to the exact-real objective of the stored inputs. Integer ceiling scores and the monotone completion evaluator add the assignment allowance in Section C.9.

All acceptance decisions use $U_1 < L_0 - \gamma$ with these enclosing endpoints. The implementation compares binary64 U_1 strictly with the round-to-nearest value of $L_0 - \gamma$. If that subtraction rounds up, every representable value strictly below it is still below the exact difference; if it rounds down, the comparison is conservative. Thus the strict floating comparison preserves the stated real inequality. Equality proceeds to the lower-witness or fallback checks. A numerical scale, integer-score, mass, or expression-range failure returns unknown before a branch query. The global reference then checks its own allowance. An inadequate reference allowance raises a failed solve; it is counted as a failed run rather than assigned a risk-change sign.

Policy	Descent	Nondecrease	Unknown	Fallback	Range fail.
Probe/root	12,901	371	1,519	1,519	0
Greedy	12,934	870	987	987	0
Uniform	12,933	858	1,000	1,000	0

Table 19: Network proposal outcomes before global fallback, aggregated over all public and synthetic configurations. Each policy tests 14,791 endpoints. Descent, nondecrease, and unknown partition those endpoints; each unknown invokes one fallback. Range failures are a subset of unknown.

Decision accuracy over complete network trajectories. Independent reevaluation of all 560 trajectories and 59,164 proposals gives exactly the same final decisions and risk intervals. The pre-fallback counts are in Table 19; the table below gives final outcomes after any fallback and baseline

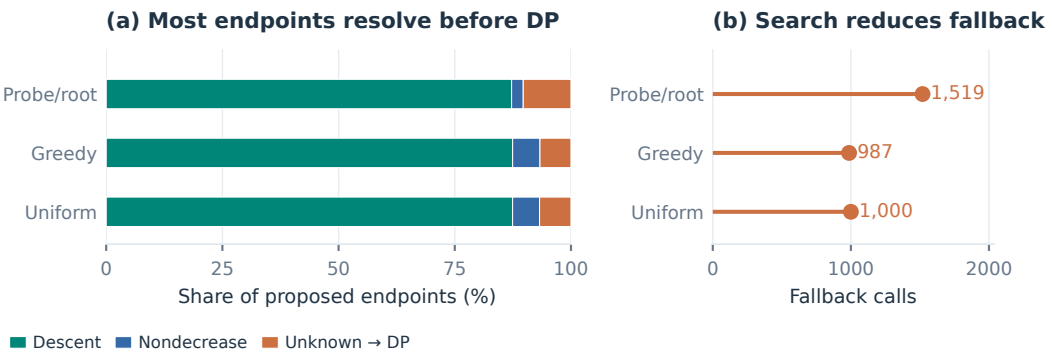


Figure 23: Network decisions before global fallback. Each policy tests the same 14,791 endpoints across 140 trajectories. Descent, nondecrease, and unknown partition the proposals; every unknown invokes DP. Greedy reduces fallback calls from 1,519 to 987 relative to probe/root, a 35.0% reduction. These counts measure solver use; Figure 7 shows the corresponding runtime tradeoff on each public-data setting.

refresh. Each row comprises 140 trajectories and 14,791 proposed endpoints. All 560 planned trajectories completed, with zero failed runs. No numerical-range failure occurs in this reevaluation.

Policy	Descent	Nondecrease	Unknown	Baseline refreshes
Direct DP	13,897	893	1	0
Probe/root	13,897	894	0	973
Greedy	13,897	894	0	987
Uniform	13,897	894	0	986

The direct-DP unknown is an Energy proposal whose padded intervals overlap; it is rejected. The shared policies resolve that proposal as nondecrease with their retained bounds. Every policy therefore follows the same accepted parameter path. All 55,588 accepted updates pass the separately saved tight reference enclosures; the closest accepted change has upper bound -7.9584×10^{-8}, below the required -10^{-8}.

Comparisons near a decision boundary. In the 6,480 matched-proposal study, the smallest observed absolute objective change is 1.6037×10^{-7}. The 140-endpoint width sweep has minimum absolute change 1.5314×10^{-5} and deliberately widens the enclosures across these separations. The production checker was also exercised on six constructed arithmetic cases: equality at the root comparison, one representable step to either side, equality at the nondecrease witness, an excessive threshold scale, and reference padding smaller than its derived error bound. The first three return unknown, descent, and unknown, respectively; the witness equality returns nondecrease. The excessive scale returns unknown with zero branch queries, and inadequate reference padding raises an explicit failure. These boundary cases are reported separately from the training counts.

E.2 Threshold geometry on regression proposals

For proposals that require additional endpoint information, candidate cover sizes one, two, and three give mean uniform-discovery query counts of 1.01, 2.21, and 3.21. The corresponding expectation bounds in Theorem 3 average 1.23, 4.15, and 9.30. These measurements relate the number of thresholds sufficient for descent to the work needed to discover them.

The geometry study contains 811 decreasing proposals from states 80 and 240 at scales $0.005, 0.5, 4$. Monotone searches on both sides of each branch minimum recover its strict sublevel interval. The construction in Section A.3 then gives a minimum cover. These quantities are computed after the policy run and provide no input to the checker. Query counts are averaged over eight policy seeds within each endpoint and then over independent data seeds; each expectation bound uses the candidate cover at the allowance-adjusted level.

E.3 Adding derivatives to shared threshold evaluations

The signed-sweep policy adds threshold subgradients to the shared value responses. On the same 6,480 proposals, it uses breadth-first midpoint search with a 256-sweep budget and depth limit 48. It decides every endpoint in at most 19 sweeps, with a mean of 3.46. Uniform branch discovery uses eight policy seeds per endpoint, with streams determined by data seed, configuration, state, scale, and policy seed. Branch queries return history-independent tail quantiles.

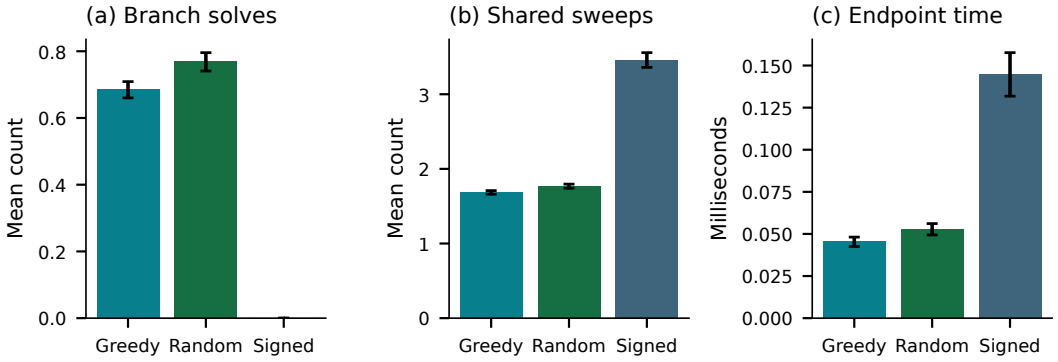


Figure 24: Effect of signed threshold responses on the same 6,480 proposals. The panels report branch solves, shared sweeps, and complete endpoint time. Uniform-discovery work is averaged over eight policy seeds per endpoint. All three policies agree with enumeration on every endpoint.

E.4 Enclosure width and decision resolution

This study holds proposed endpoints fixed to isolate the effect of numerical width on the sign decision. It uses deletion budget ten, tail mass 0.9, free retention, and state 80. Seven scales $0.005, 0.015, 0.05, 0.15, 0.5, 1.5, 4$ and twenty seeds give 140 endpoints: 75 decrease the exact risk and 65 do not. Each is tested at requested radii $r = 0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-3}, 10^{-2}$, with $\gamma = 0$, giving 840 comparisons. The current-value half-width is $\max\{r, 10^{-9}\}$, and the branch and response allowance is $\max\{r, 2 \cdot 10^{-11}(1 + \max_i \widehat{\ell}_i + |\widehat{R}|)\}$. Thus $r = 0$ retains the arithmetic allowance rather than representing exact arithmetic.

Requested r	Descent	Nondecrease	Unknown	Resolved (%)	Mean queries	Errors
0	75	65	0	100.0	1.41	0
10^{-8}	75	65	0	100.0	1.41	0
10^{-6}	75	65	0	100.0	1.42	0
10^{-4}	67	61	12	91.4	1.44	0
10^{-3}	32	57	51	63.6	1.66	0
10^{-2}	0	43	97	30.7	4.86	0

Table 20: Decision resolution at the same 140 endpoints, with zero required decrease. Each row contains seven update scales on each of twenty seeds. The baseline radius is $\max(r, 10^{-9})$; branch evaluations also retain their data-dependent roundoff floor. Errors compare returned signs with the stored exact-risk differences. These records contain no fallback or time measurements.

Table 20 separates incorrect decisions from unresolved ones. No returned sign is wrong at any radius. Unresolved comparisons increase to 12, 51, and 97 at $10^{-4}, 10^{-3}, 10^{-2}$, respectively. At the largest radius, all 75 actual decreases remain unresolved, along with 22 nondecreases. The remaining 43 nondecreases still have separating bounds. Within every endpoint, neither the unknown indicator nor the number of queries decreases as r increases. Mean queries rise from 1.41 to 4.86. This is a loss of decision resolution while retaining valid definite decisions, consistent with Equation (5).

The fixed-endpoint records measure queries and resolution. They contain no fallback calls or elapsed times. The training study in Section D.4 measures those downstream costs on its own matching trajectories, where the varied radius pads the DP reference and does not change the tree's arithmetic allowance. The two studies therefore test distinct parts of the mechanism.

E.5 Remaining objective gap

For the linear small-family experiment, Figure 25 compares the six acceptance policies with the convex optimum bound from Section C.7.

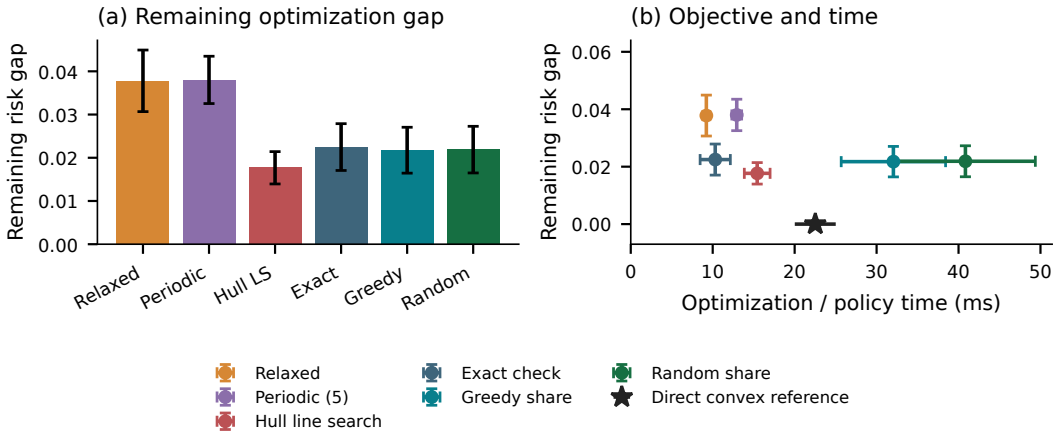


Figure 25: Objective progress in the linear small-family experiment. Left: mean remaining risk above the optimum lower bound computed with rational arithmetic. Right: achieved objective gap versus complete policy time; the star denotes direct convex optimization. Error bars aggregate data seeds.

E.6 Query budgets and risk configurations

Figures 26 to 28 group the matched proposal grid by query budget, path position, and risk configuration.

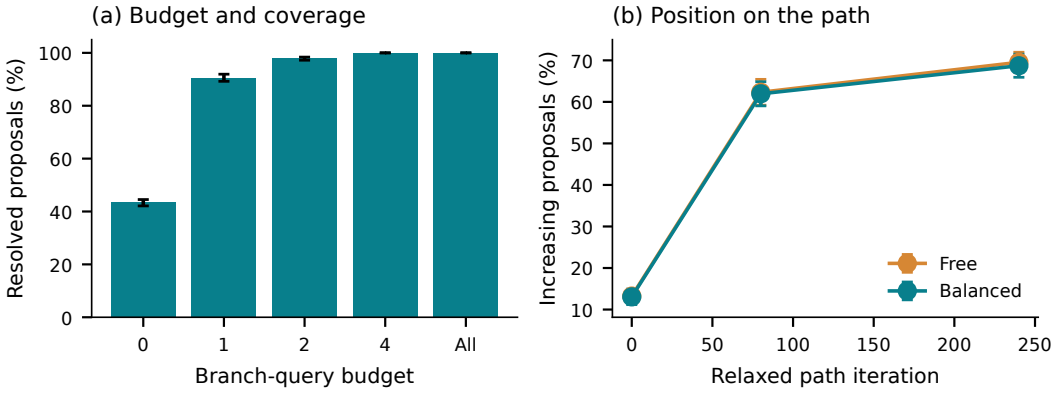


Figure 26: Query-budget and path-position ablations. A smaller query cap reduces both decision coverage and branch work. The path-position panel groups proposals by the number of preceding relaxed updates, using the same configurations and step scales.

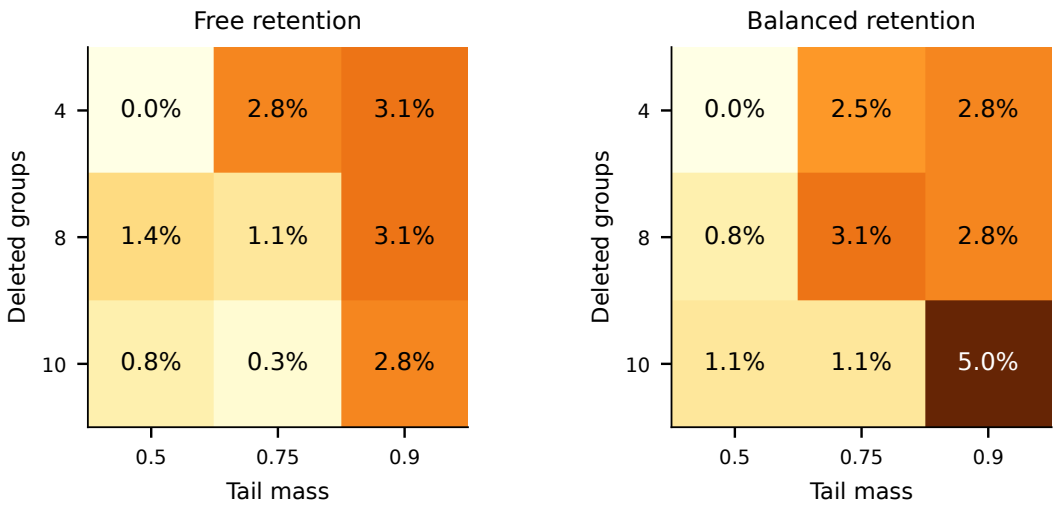


Figure 27: Percentage of proposals that decrease the relaxed objective while increasing the discrete objective, by deletion budget, tail mass, and retention constraint. Each cell includes all evaluation seeds, path positions, and step scales for that configuration.

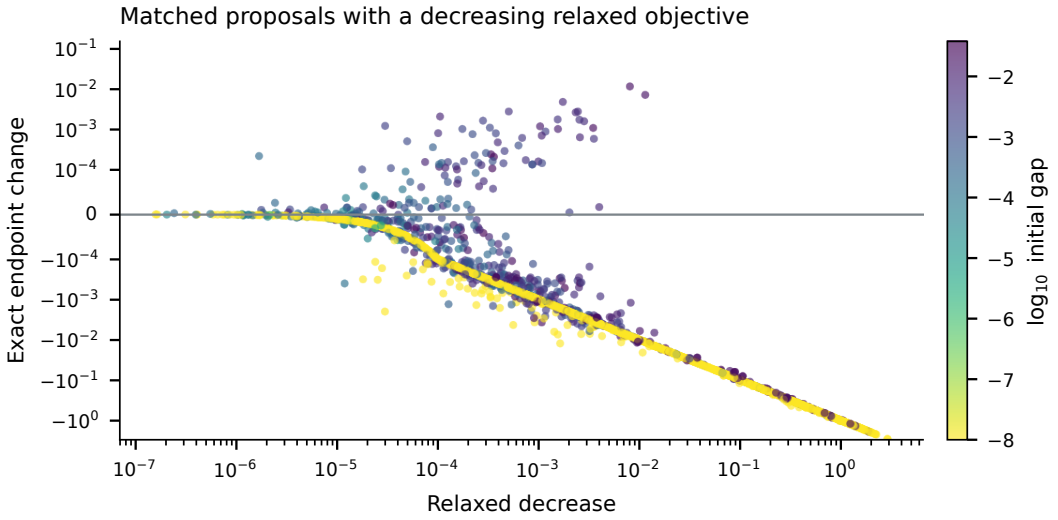


Figure 28: Discrete endpoint changes among proposals that decrease the relaxed objective. Points above zero increase the discrete risk. Color indicates the initial gap between the objectives. The horizontal axis is logarithmic; the vertical axis is symmetric logarithmic near zero.