BAG DESIGN FOR MODEL EVALUATION FROM AGGREGATE LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Aggregate labels provide responses for groups of instances, or bags, when individual outcomes are unavailable. Existing bag-curation methods target gradient recovery or prediction quality, while evaluating a fixed model requires understanding how grouping affects uncertainty about its ranking performance. Identical group means can support different values of the area under the receiver operating characteristic curve (AUC). We formulate bag design before feedback as minimizing the range of AUC values compatible with exact population group means. When each instance has its own response probability, we derive an exact width formula that sums contributions from score ranks within bags. For distinct scores and equal-capacity bags, adjacent-rank grouping minimizes worst-case width at a fixed overall response probability, provided its product with bag size is an integer. When instances with identical observed features share a response probability, their occurrences across bags couple the constraints. Using specified reference probabilities, we compute the coupled AUC endpoints by linear programming and search for width-reducing swaps that preserve bag sizes. Keeping each repeated state within one bag recovers the sum obtained by treating instance probabilities separately. On twelve-instance blocks, width-minimizing assignments that spread repeated states across bags reduce the median coupled-to-separable width ratio to 0.104, demonstrating that evaluation-width minimization and training-oriented curation can select different bags. In the separate-probability setting, adjacent-rank grouping yields narrower intervals than balanced random assignment across click-log, census and network-traffic workloads. On the click log at a fixed feedback budget, its interval width is 2.4×10^{-4} versus a balanced-random median of 0.992. These results support designing aggregate-label groups around the performance measure to be evaluated.

1 INTRODUCTION

Evaluating a fixed ranking model becomes difficult when labels are released only as group averages. A platform may report the response rate of a group of impressions while withholding individual clicks. Such groups, or *bags*, reveal how much response probability they contain without identifying which score ranks carry it. The area under the receiver operating characteristic curve (AUC) measures how often a positive draw ranks above a negative one (Agarwal et al., 2005). Moving response probability between high and low ranks within a bag can preserve its mean while changing this performance measure. The same aggregate feedback can therefore support different AUC values for the same model.

Learning from label proportions uses aggregate responses to train models (Quadrianto et al., 2009; Ardehaly & Culotta, 2017; Busa-Fekete et al., 2023). Bag-curation methods improve this supervision by grouping instances using features or predicted responses (Chen et al., 2023; Javanmard et al., 2024b; Agarwal et al., 2025). For a fixed scorer, the PriorBoost partitioning step and the instance-level regression design of Agarwal et al. (2025) both group consecutive score ranks. These constructions motivate a separate evaluation question: which grouping leaves the least uncertainty about the AUC of an already trained model?

Partial identification provides the relevant notion of uncertainty by considering all target values compatible with the observations (Molinari, 2020). In weak-supervision evaluation, Polo et al.

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.

(2024) bound performance metrics under given marginal constraints. Design can also change the constraints themselves, as in the instrument partitions that Schweisthal et al. (2025) learn to tighten treatment-effect bounds. Here, we study the grouping of instances as a design variable for AUC evaluation.

We define the *sharp AUC width* as the difference between the largest and smallest attainable AUC over response probabilities compatible with exact bag means. The curator chooses equal-sized bags before feedback is released, using the same instance pool and number of reported means across designs (Section 3). Minimizing this width makes the released information more informative about the fixed model.

We first allow each instance to have its own response probability. At a fixed overall response rate, or prevalence, AUC admits an affine representation and its sharp width decomposes into within-bag contributions (Proposition 1). Grouping adjacent score ranks restricts how far response probability can move within each bag. For distinct scores and a common response probability across instances, we derive exact comparisons with balanced random assignment and round-robin assignment, which spreads consecutive ranks across bags. When bag capacity times prevalence is an integer, adjacent-rank grouping also minimizes the worst-case width over response probabilities at that prevalence. Thus the existing training-oriented partition has a precise evaluation justification in this setting.

A second setting arises when instances with identical observed features share a response probability. Occurrences of the same state in different bags then couple their mean constraints. We compute sharp AUC endpoints with linear programs that enforce these constraints jointly (Section 5). To improve a partition, we search over swaps that preserve bag capacities, using specified reference probabilities to evaluate the means and width of each candidate before feedback. This shared-probability setting can separate the grouping preferred by the PriorBoost objective from the grouping preferred by AUC width.

Experiments evaluate these two settings separately. With a separate probability for each instance, adjacent-rank grouping narrows AUC intervals across click-log, census and network-traffic workloads. For the click log with $4{,}096$ reported bag means, its width is approximately 2.4×10^{-4}, compared with a balanced-random median of 0.992 (Table 1). For repeated states, exhaustive enumeration on 20 twelve-row blocks from Criteo and Adult finds disjoint sets of width-minimizing and training-risk-minimizing partitions (Section 6.4). The former result measures the information retained by rank-based grouping, while the latter motivates optimizing the evaluation objective directly when probabilities are shared.

Our contributions are as follows.

- We formulate bag design as minimizing the sharp AUC width remaining after exact aggregate feedback (Section 3).
- We derive the width decomposition for separate instance probabilities and establish the conditional worst-case optimality of adjacent-rank grouping (Theorems 2 and 4).
- We develop joint endpoint programs and capacity-preserving search for repeated states, and evaluate how both probability settings affect the choice of bags (Sections 5 and 6).

2 Related Work

Learning from label proportions. LLP trains instance predictors from group responses using aggregate losses (Musicant et al., 2007) or mean maps and latent labels constrained by the reported bag proportions (Quadrianto et al., 2009; Yu et al., 2013).

Learning guarantees and evaluation benchmarks. LLP theory studies learnability (Yu et al., 2014; Fish & Reyzin, 2017), while benchmarks show sensitivity to bag generation (Brahmbhatt et al., 2024; Franco et al., 2023) and loss comparisons distinguish instance-level from bag-level training (Javanmard et al., 2024a). Label-inference analysis quantifies disclosure from aggregates (Busa-Fekete et al., 2024), whereas we bound a fixed scorer's AUC under exact bag means.

Bag construction and group design. Bag-construction methods improve prediction by combining observed bags or augmenting them through overlap (Saket et al., 2022; Asanomi et al., 2023). Other approaches use smaller bags or perturb proportions in large bags to improve aggregate-supervised

training (Baručić & Kybic, 2022; Kubo et al., 2024). Direct curation groups common-feature instances to recover gradients (Chen et al., 2023), whereas PriorBoost groups predicted responses and updates the model iteratively (Javanmard et al., 2024b).

Partial identification and set-valued estimands. Partial identification bounds target values over distributions compatible with observations, including ecological aggregates (Duncan & Davis, 1953; Jiang et al., 2020) and broader incomplete-data restrictions (Molinari, 2020; Moon, 2026). Applications optimise under measurement error or contamination (Finkelstein et al., 2021; Nafshi & Makar, 2024), while weak-supervision evaluation bounds metrics from observed marginals (Polo et al., 2024). Conditioning on the bag index, that Fréchet form is our separable width at fixed prevalence. Learned instrument partitions additionally tighten treatment-effect bounds (Schweisthal et al., 2025) by changing the representation used to form constraints. Repeated observable states instead tie one probability to several bag means, so the resulting equalities lie outside any bound formed from bag marginals alone. Here the partition is the design variable rather than an input, and Appendix F compares the feasible sets.

Ranking evaluation and ordered partitions. AUC evaluates positive-negative score comparisons through rank statistics (Mann & Whitney, 1947; Hanley & McNeil, 1982), while generalisation theory quantifies labelled-sample error (Agarwal et al., 2005). At fixed prevalence, the affine representation proved here supplies coefficients determined by score ranks. Contiguous optimal partitions in one dimension go back to Fisher (Fisher, 1958) and underpin interval clustering (Wang & Song, 2011; Nielsen & Nock, 2014) with objective-specific accelerations (Grønlund et al., 2017). On fixed scores, score-contiguous grouping is the partition that the PriorBoost clustering step and the bagging of Agarwal et al. (2025) both select. We show that with distinct scores at integer budgets it also minimises the largest AUC width for separate rows and that repeated states break this agreement.

3 Problem Formulation

We choose bags to preserve information about a fixed model's AUC when feedback reveals exact population group means. Section 3.1 specifies the design inputs and target, and Section 3.2 defines the AUC range left by this feedback. We first allow a separate response probability for each row, then impose a shared probability on repeated states in Section 5.

3.1 Design Inputs and Evaluation Target. The evaluation pool contains N instances, called rows, with fixed scores $s_i \in \mathbb{R}$. A balanced partition $B = (B_1, \dots, B_m)$ has m non-overlapping bags of capacity n, so $N = mn$. The curator knows the scores and observable feature identities. Pointwise width minimisation additionally takes a specified reference probability vector before design. In contrast, the canonical minimax rule in Section 4.3 uses score ranks and a fixed prevalence. The scorer, rows, bag count and capacities remain fixed as membership changes. After choosing B, the evaluator receives the bag means and computes an AUC interval.

Let $\rho_i \in [0,1]$ be row i's positive-response probability and call ρ a latent world. Its prevalence is $\pi = N^{-1}\sum_i \rho_i$. We compare worlds in $\mathcal{R}_\pi = \{\rho \in [0,1]^N : \mathbf{1}^\top \rho = N\pi\}$ for $0 < \pi < 1$. For pointwise design, π is determined by the reference vector and remains common across candidate partitions. Define the score comparison $h(s_i, s_k) = \mathbb{1}\{s_i > s_k\} + \frac{1}{2}\mathbb{1}\{s_i = s_k\}$. The soft-label pairwise AUC is

$$\mathrm{AUC}(s, \rho) = \frac{\sum_{i=1}^N \sum_{k=1}^N \rho_i(1-\rho_k)h(s_i, s_k)}{N^2\pi(1-\pi)}. \tag{1}$$

This compares an independent positive and negative draw from the finite row pool, weighting their indices by ρ_i and $1-\rho_k$, respectively. The indices may coincide, in which case the score comparison receives half credit. For binary ρ, the expression reduces to empirical AUC with half credit for score ties. Fixing π fixes its denominator.

3.2 Compatible Worlds and Sharp Width. Bag j releases the population mean $\mu_j^{\mathrm{rel}} = n^{-1}\sum_{i \in B_j} \rho_i$. Write $b_j^{\mathrm{rel}} = n\mu_j^{\mathrm{rel}}$ for its total probability mass, which lies in $[0, n]$ and need not be an integer. This differs from an empirical count of realised binary outcomes. The exact-mean

compatible set is

$$\mathcal{I}_{\text{mean}}(B, b) = \left\{ \widetilde{\rho} \in \mathcal{R}_\pi : \sum_{i \in B_j} \widetilde{\rho}_i = b_j \text{ for all } j \right\}. \tag{2}$$

For feasible b, let $L(B \mid b)$ and $U(B \mid b)$ be the minimum and maximum AUC on this set. Their difference $\mathsf{W}(B \mid b) = U(B \mid b) - L(B \mid b)$ is the sharp identification width: both endpoints are attainable, and zero width means that all compatible worlds agree on AUC. This is uncertainty from incomplete identification under exact means, not a sampling confidence interval. Appendices A.2 and A.3 extend the formulation to stochastic releases and explain when feedback can make assignment consequential.

4 Exact Widths and Design Objectives

We derive the exact width under population mean feedback and then compare canonical assignments. The separate-row model assigns one free probability coordinate to every row, including rows with identical observed features. This is a modelling choice, not a claim of statistical independence.

4.1 Reference Worlds and Design Objectives. The probability masses in (2) become available after design. To choose bags beforehand, we must specify how candidate releases are compared. Throughout, b_j denotes probability mass, m counts released bag means, and a search budget counts candidate evaluations.

Two design objectives. The released budgets become available only after B has been chosen. To compare candidate partitions before feedback, let $\rho^0 \in \mathcal{R}_\pi$ be a specified reference world. Each candidate induces its own reference budgets $b_j^{\text{ref}}(B) = \sum_{i \in B_j} \rho_i^0$ from this common input. Let $\mathcal{B}_{m,n}$ denote the class of equal-capacity partitions. We abbreviate $\mathsf{W}(B \mid b^{\text{ref}}(B))$ as $\mathsf{W}(B \mid \rho^0)$ when the reference world is the argument. We consider a pointwise objective and a minimax objective: $\min_{B \in \mathcal{B}_{m,n}} \mathsf{W}(B \mid \rho^0)$ and $\min_{B \in \mathcal{B}_{m,n}} \sup_{\rho^0 \in \mathcal{R}_\pi} \mathsf{W}(B \mid \rho^0)$, where the supremum ranges over all reference worlds at the fixed prevalence. The pointwise objective compares partitions for one specified world and is the objective of the shared-state search in Section 5. The minimax objective minimises the largest width over the fixed-prevalence class and yields the canonical guarantee below. Each candidate recomputes its own $b^{\text{ref}}(B)$ from the same ρ^0. After deployment, the evaluator uses b^{rel} to form the actual interval for the selected partition. We next derive the width needed to evaluate both objectives.

4.2 Exact Width Geometry. We evaluate the pointwise and minimax objectives by rewriting pairwise AUC through score ranks and then separating the bag constraints.

Affine reduction. Only the order of the fixed scores enters the AUC comparison. To express this order as a coefficient for each response probability, we use the normalised rank contrast: $G_i = N^{-1} \sum_{k=1}^{N} [\mathbb{1}\{s_i > s_k\} - \mathbb{1}\{s_i < s_k\}]$, where G_i is the fraction of rows scored below row i minus the fraction scored above it. A high-ranked row has positive contrast, and a row with equal numbers above and below has zero contrast. For distinct scores, let r_i be the ascending rank of row i. Then $G_i = (2r_i - N - 1)/N$. The symmetry $h(s_i, s_k) + h(s_k, s_i) = 1$ makes the quadratic term in (1) equal to $N^2\pi^2/2$. At fixed prevalence it is constant, leaving the following affine dependence on ρ.

Proposition 1 (Affine reduction of pairwise AUC)**.** *At fixed prevalence, the target in* (1) *satisfies* $\text{AUC}(s, \rho) = \frac{1}{2} + \left(\sum_{i=1}^{N} \rho_i G_i \right) / (2N\pi(1 - \pi))$.

AUC is therefore a constant plus a weighted sum of response probabilities. The affine reduction and bagwise endpoint proofs appear in Appendices A.1 and A.4.

Separable width. Suppose each row has its own response probability, so its coordinate appears in exactly one bag constraint. We can then optimise each bag independently. For a bag with contrast vector $g \in \mathbb{R}^n$ and budget $b_j \in [0, n]$, let $\Delta(g, b_j)$ denote the range of $g^\top v$ over $\{v \in [0,1]^n : \mathbf{1}^\top v = b_j\}$. Here v is the vector of candidate probabilities within this single bag. Writing $k = \lfloor b_j \rfloor$ and $f = b_j - k$, the maximum assigns unit probability to the k largest coefficients and mass f to the next, while the minimum fills from the other end (Bertsimas & Tsitsiklis, 1997). At $b_j = 0$ or n, the allocation is fixed and the range is zero. The next theorem combines the single-bag ranges into the exact AUC width.

Theorem 2 (Sharp width as within-bag rank transport)**.** *Fix the AUC target and prevalence under exact-mean feedback. Suppose every row has its own probability coordinate. For any balanced partition B and feasible budget vector b, the sharp width is $\mathsf{W}(B \mid b) = \sum_{j=1}^{m} \Delta(G_{B_j}, b_j)/\{2N\pi(1-\pi)\}$, where G_{B_j} collects the rank contrasts of rows in bag j. Both endpoints are attained by bagwise continuous-knapsack solutions.*

The rank-transport interpretation describes movement of probability between low and high ranks within a bag. Grouping nearby ranks restricts this movement while preserving a bag mean. When the reference world is a realised outcome vector, every bag budget is an integer count, so both knapsack optima are binary and the width above is also the exact AUC range over the labellings that reproduce those counts. This binary reference special case does not identify a stochastic world from one count release.

4.3 Canonical Assignments and Optimality. We use the width decomposition to compare ways of distributing score ranks across bags at the same feedback budget. The canonical class has separate row probabilities and distinct scores under fixed-prevalence exact-mean feedback with equal capacities. At the flat reference world $\rho_i^0 = \pi$, every bag has budget $n\pi$. Score-contiguous assignment B^{sc} groups consecutive ascending ranks. Round-robin assignment B^{rr} sends consecutive ranks to successive bags cyclically. The random design is uniform over balanced partitions. Define $\Psi(b, n) = b(n-b) - f(1-f)$ with $f = b - \lfloor b \rfloor$. The fractional correction accounts for the one partially filled coordinate in a knapsack endpoint.

Proposition 3 (Canonical three-design spectrum)**.** *In the canonical design class at the flat reference world the three widths are $\mathsf{W}(B^{\mathrm{sc}} \mid \pi\mathbf{1}) = m\Psi(n\pi, n)/\{N^2\pi(1-\pi)\}$, $\mathsf{W}(B^{\mathrm{rr}} \mid \pi\mathbf{1}) = m\mathsf{W}(B^{\mathrm{sc}} \mid \pi\mathbf{1})$ and $\mathbb{E}_B\mathsf{W}(B \mid \pi\mathbf{1}) = \{(N+1)/(n+1)\}\mathsf{W}(B^{\mathrm{sc}} \mid \pi\mathbf{1})$, where the expectation is over uniform balanced partitions. If $n\pi$ is an integer, the contiguous width reduces to $1/m$ and the round-robin width to 1.*

Round-robin therefore multiplies the contiguous width by the bag count and random assignment by $(N+1)/(n+1)$ in expectation. This expectation over random partitions is distinct from the empirical random-assignment median reported later. We next establish when contiguous grouping minimises width over all partitions.

Theorem 4 (Flat-world and canonical minimax optimality)**.** *In the canonical design class, score-contiguous assignment minimises width at the flat world. If in addition $n\pi$ is an integer, it solves the minimax problem over fixed-prevalence reference worlds with value $1/m$. The flat world then attains the largest width for B^{sc}.*

The proofs appear in Appendices A.6 and A.7. For every $n\pi$ in the same canonical class, the latter also gives $\sup_{\rho^0 \in \mathcal{R}_\pi} \mathsf{W}(B^{\mathrm{sc}} \mid \rho^0) = m\Psi(n\pi, n)/\{N^2\pi(1-\pi)\} \le 1/m$. Thus the $1/m$ upper bound holds throughout this canonical class. Flat-world optimality requires no integrality assumption, whereas the stated minimax equality requires integer $n\pi$. The theorem does not supply an optimum for arbitrary pointwise worlds or shared-state constraints. In the separate-row model, width vanishes exactly when each bag has mass 0 or n, or constant rank contrasts (Appendix A.8).

5 Repeated States and Direct Width Minimisation

We now impose a shared probability on repeated observable states. A state appearing in several bags couples their constraints, so its probability must satisfy all released means simultaneously. We compute the resulting endpoints jointly, then use them to compare capacity-preserving swaps at a fixed reference world.

5.1 Coupled Endpoints. Let $\mathcal{A} = \{1, \dots, K\}$ index distinct observable states, called atoms, and let $a(i)$ denote the atom of row i. A state key records the selected scorer inputs: the small-block experiments use raw numeric keys, while the full-window extension uses complete preprocessed feature vectors (Appendices B.4 and C.12). Rows in one atom share a probability $u_a \in [0, 1]$, so $\rho_i = u_{a(i)}$, and receive the same score. Score ties alone do not merge different states. This equality of probabilities is an additional model constraint.

To express its effect, define the $m \times K$ bag–atom incidence matrix $(A_B)_{ja} = |\{i \in B_j : a(i) = a\}|$. Each entry counts occurrences, including multiplicities, rather than merely indicating presence. The

compatible atom worlds satisfy $A_B\widetilde{u} = b$ and $0 \le \widetilde{u} \le 1$. Because bags partition all rows, summing these equations gives the fixed total mass $\sum_j b_j = N\pi$. Likewise, aggregate the rank contrasts as $c_a = \sum_{i:a(i)=a} G_i$. Proposition 1 then reduces the AUC endpoints to linear programs in the shared variables (Bertsimas & Tsitsiklis, 1997).

Proposition 5 (Shared atoms yield exact coupled endpoints)**.** *Fix $0 < \pi < 1$ under exact-mean feedback. For an equal-capacity assignment B and feasible mass vector b, let $V^-(B \mid b)$ and $V^+(B \mid b)$ be the minimum and maximum of $c^\top \widetilde{u}$ subject to $A_B\widetilde{u} = b$ and $0 \le \widetilde{u} \le 1$. Then $L(B \mid b) = \frac{1}{2} + V^-(B \mid b)/\{2N\pi(1-\pi)\}$, and $U(B \mid b)$ uses V^+ in the same expression. Hence $\mathsf{W}(B \mid b) = \{V^+(B \mid b) - V^-(B \mid b)\}/\{2N\pi(1-\pi)\}$. Both optima are attained on the nonempty closed and bounded feasible set.*

The coupled width is at most the separable sum obtained by allowing every occurrence its own coordinate. Singleton atoms recover Theorem 2. Keeping every atom within one bag also gives equality, since its equal-score occurrences have identical rank coefficients. Spreading an atom across bags can further restrict the compatible worlds, although it need not reduce width for every assignment. Appendix A.9 proves the endpoint result, and the coupling comparison in Appendix C.8 measures this effect.

For pointwise design, specify reference atom probabilities u^0 before search and set $\pi = N^{-1}\sum_i u^0_{a(i)}$. Every candidate uses $b^{\mathrm{ref}}(B) = A_B u^0$, which guarantees feasibility at this common prevalence. The experiments use smoothed evaluation-shard response rates for u^0 (Appendix B.4), so their search comparisons are conditional on these reference inputs. The separate-row experiments instead use the fixed evaluation outcome vector. After feedback, the evaluator substitutes b^{rel} and checks feasibility under the chosen state model. Empirical counts can violate the shared-state exact-mean constraints (Appendix E.5).

Risk and width objectives. The PriorBoost-style partition criterion is $R(B) = \sum_j \sum_{i \in B_j} (s_i - \overline{s}_j)^2$, where $\overline{s}_j = n^{-1}\sum_{i \in B_j} s_i$ (Javanmard et al., 2024b). Here "risk" denotes within-bag squared variation of fixed scores, not the test risk of a retrained model. It ignores cross-bag state sharing, whereas width uses that sharing through A_B. The twelve-row construction in Appendix A.10 exhibits disjoint risk- and width-minimiser sets.

5.2 A Width-Directed Search. The endpoint programs solve a fixed assignment exactly. Choosing that assignment is a separate discrete optimisation problem, which we approach by local search. Given the scores, atom map, capacities and u^0, the search minimises $\mathsf{W}(B \mid u^0) := \mathsf{W}(B \mid A_B u^0)$. It starts from the PriorBoost-style primary partition and balanced random partitions. From each start, it evaluates every cross-bag row swap, selects the largest width decrease above the comparison tolerance, and resolves numerical ties by lexicographic order. Exchanging two rows preserves capacity, and strict descent prevents cycles.

A trajectory stops when no eligible swap remains or its step limit is reached. Only the first condition certifies local optimality within the tolerance. The algorithm returns the best final assignment across starts, with its stopping status. Global optimality is assessed separately by enumeration on small blocks. Algorithm 1 specifies initialisation, traversal and output, and Appendix A.11 accounts for the paired LP cost across all starts.

6 Experiments

We test two consequences of the theory: nearby ranks reduce ambiguity with separate row probabilities, and cross-bag state sharing can make direct width minimisation preferable to the squared-score criterion. We then separate the contributions of descent, starting assignments and candidate-evaluation budget.

6.1 Experimental Setup. We use Criteo click logs (Criteo AI Lab, 2015), Adult income records (Becker & Kohavi, 1996) and UNSW-NB15 traffic (Moustafa & Slay, 2015). Logistic scorers and preprocessing are fitted on disjoint leading blocks and remain fixed (Pedregosa et al., 2011). Table 7 separates the large row-coordinate windows from the shared-state block and window studies. The former use realised binary outcomes as the reference, so their widths also bound AUC across label vectors reproducing those counts. The latter use Jeffreys-smoothed evaluation-shard rates as reference

Table 1: Separate row probabilities at 4,096 bags per workload, using realised outcome references. Random is the median of ten balanced assignments; other columns are deterministic. Feature bins sort the same rows by one selected feature. Criteo and Adult use shard 0. Table 18 reports random-median uncertainty, and Table 10 lists Criteo replicates.

Workload	Rows	Capacity	Contiguous	Feature	Random	R-robin
Criteo	1,048,576	256	2.40×10^{-4}	0.894	0.992	0.996
Adult	16,384	4	1.41×10^{-4}	0.431	0.629	0.796
UNSW-NB15	524,288	128	4.72×10^{-5}	0.265	0.954	0.961

probabilities and evaluate the induced population means. They are pointwise design comparisons conditional on these label-derived inputs.

The final shared-state grid has ten capacity-four blocks at each of 12, 16, 24 and 32 rows per workload, split equally between two shards. Features determine atom eligibility, ordering and block membership; reference probabilities then determine candidate widths. Each selected atom contributes two rows. Repeated states cover less than 1% of Criteo rows but about 70% of Adult rows (Table 8). Enriched blocks isolate the shared-state mechanism, whose availability across a full workload depends on this frequency.

The risk reference minimises the within-bag squared variation of fixed scores defined in Section 5.1. At twelve rows we choose the narrowest of all risk minimisers; at larger sizes we use the stable score-contiguous representative. Enumeration confirms uniqueness at sixteen rows. Smaller AUC identification width is better. Paired reduction is $(W_{\mathrm{risk}} - W_{\mathrm{search}})/W_{\mathrm{risk}}$ within each block. Reported bootstrap intervals describe block or random-assignment variability, as specified in each display, and are separate from the identification interval.

6.2 Adjacent Ranks Retain More AUC Information. Table 1 holds the feedback count at 4,096 bags within each workload. On Criteo, contiguous grouping leaves width about 2.4×10^{-4} versus a random median of 0.992, moving from almost the full AUC range to a narrow interval through membership alone. Theorem 2 explains this difference: a contiguous bag permits probability to move only between nearby ranks. Equal-capacity bins of one observed feature improve on random grouping but remain wider than contiguous bags. That feature-based grouping is less aligned with the AUC rank contrast (Table 21); this comparison concerns fixed-model evaluation.

Figure 1(a) shows how this comparison changes with capacity on Adult shard 0 and UNSW-NB15. Contiguous width stays below every random width across both Adult shards and all network settings (Tables 24 and 25). At network capacity 32, tied scores reverse the random–round-robin ordering while contiguous grouping retains its advantage. This illustrates how ties can alter one part of the canonical spectrum. Table 30 separately compares flat-world predictions with the non-flat observed worlds.

6.3 Controls Identify the Role of Ranks and Feedback. To remove rank alignment, we form contiguous bags after sorting independently generated random scores, then evaluate the original scorer. This control does not preserve the original score ties. Its paired difference from balanced random assignment is small relative to the contiguous-to-random gap and has mixed signs (Table 26); it shows no systematic gain comparable to using the scorer's own order. A separate implementation check replaces means with witness-blind feedback, whose law is independent of the probability world. All 22 Criteo assignments then leave width one (Table 34), as Proposition 6 implies. Finally, the separate-row widths agree with the Fréchet-type bound of Polo et al. (2024) on the same assignments. Thus these gains arise from informative grouping under the stated constraints (Appendix C.11).

6.4 Shared States Separate the Two Design Objectives. We next test whether minimising squared score variation also minimises width when states repeat. Table 2 compares complete minimiser sets and distinguishes global minima from search outputs. The sets are disjoint on all tested twelve- and sixteen-row final-grid blocks. Search attains every twelve-row minimum but only 12 of the 20 sixteen-row minima. Enumeration therefore establishes objective separation, while also revealing residual search error at the larger size.

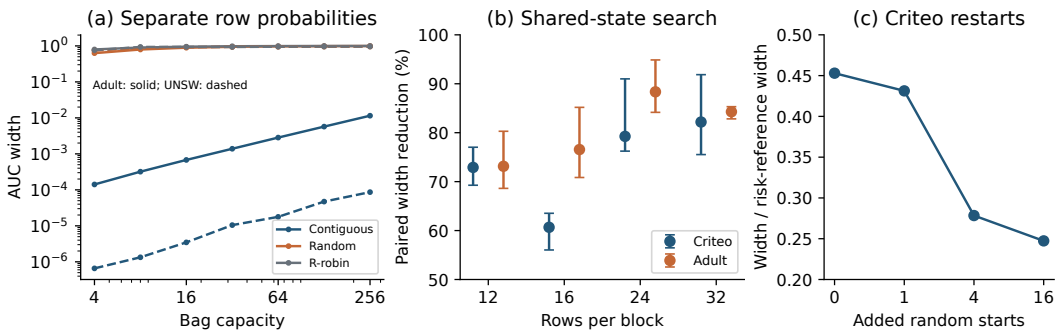

Figure 1: Distinct sources of width reduction. (a) Separate-row capacity trends for Adult shard 0 and UNSW-NB15. (b) Shared-state median paired reduction with 95% descriptive block-bootstrap intervals: ten blocks per workload and size, 10,000 resamples. These are medians of paired reductions, not ratios of median widths. (c) Criteo restart ablation over the same forty blocks, reporting median final width divided by its block's risk-reference width.

Table 2: Global checks on the final shared-state grid, ten blocks per workload and size. Enumeration covers 5,775 partitions at 12 rows and 2,627,625 at 16 rows per block. Risk and global columns are median widths. Intersection counts blocks sharing a risk and width minimiser; Hit counts search attaining the global minimum. The ten separate Criteo calibration blocks also attain their minima (Table 12).

Workload	Rows	Best risk	Global min.	Intersection	Hit
Criteo	12	0.2014	0.05362	0/10	10/10
Adult	12	0.1778	0.04475	0/10	10/10
Criteo	16	0.1481	0.05347	0/10	4/10
Adult	16	0.1656	0.03149	0/10	8/10

Figure 1(b) reports paired reductions, with absolute widths in Table 3. Search improves on the risk reference in all forty blocks of each workload. The coupled-to-separable ratio at the twelve-row optima has median 0.104 (Table 19), while contiguous and risk-based assignments retain ratio one by keeping each atom within a bag. This connects the improvement to the cross-bag equalities in Proposition 5.

Table 3: Shared-state final-grid results: ten capacity-four blocks per workload and size, five per shard, with label-derived reference probabilities. Risk and Search are marginal median widths; reduction is the median paired percentage, with a 95% descriptive interval from 10,000 block-bootstrap resamples. The risk reference is best among all minimisers at 12 rows and the stable contiguous representative otherwise.

Workload	Rows	Risk	Search	Reduction (%)	95% interval
Criteo	12	0.2014	0.0536	72.9	[69.3, 77.0]
Criteo	16	0.1481	0.0590	60.7	[56.0, 63.5]
Criteo	24	0.0993	0.0208	79.2	[76.2, 91.0]
Criteo	32	0.0738	0.0132	82.2	[75.5, 91.9]
Adult	12	0.1778	0.0447	73.1	[68.6, 80.3]
Adult	16	0.1656	0.0342	76.6	[70.8, 85.2]
Adult	24	0.1039	0.0120	88.4	[84.2, 94.9]
Adult	32	0.0780	0.0128	84.3	[82.8, 85.3]

6.5 Search Gains, Initialisation and Computational Cost. Search beats the best unoptimised member of its identical starting set in every block, establishing a benefit from descent. Additional starts improve Criteo results further, as Figure 1(c) shows. Candidate-count-matched random search supplies a separate control: the method wins or ties at sixteen rows and wins at both larger final-grid sizes (Table 28). Matching width calls controls evaluation volume, not wall-clock time. Table 4 shows the rising cost with block size and confirms that all 680 Criteo trajectories reach a local pair-swap stop. These results support both targeted descent and multiple starts.

Table 4: Criteo search cost by scale. Evaluations are median candidate-width calls per block summed over all seventeen starts; the original evaluator uses two LPs per call. Moves are per trajectory. All local stops found no improving pair swap before the 100-step cap. These counts do not match wall-clock time.

Rows	Blocks	Starts	Evaluations	Mean moves	Max moves	Local stops
12	10	17	2,297	1.85	4	170/170
16	10	17	6,401	2.87	5	170/170
24	10	17	23,897	5.08	8	170/170
32	10	17	62,737	7.11	11	170/170

To examine larger blocks, sampled first-improvement descent uses a 10,000-proposal budget on Adult blocks of 64–256 rows. Every block improves, while the median reduction decreases across these sizes (Table 33). The pattern is consistent with a fixed budget covering less of a growing search space; the experiment also changes block composition. At full-window scale, a separate family of 34 structured proposals improves on the risk reference at every tested capacity (Table 23). At capacity 256, the relative gains are about 1.4% on Adult and 1.0% on UNSW-NB15. These are best-found improvements under label-derived reference probabilities.

Table 5: Distinct scale extensions under label-derived references. Block rows give medians over five capacity-four Adult blocks using 10,000 sampled proposals. Window rows give single fixed-window results at capacity 256 using 34 structured proposals. Reductions use paired unrounded widths.

Setting	Rows	Risk width	Search width	Reduction (%)
Adult blocks	64	0.03687	4.61×10^{-3}	86.2
Adult blocks	128	0.01875	4.18×10^{-3}	77.0
Adult blocks	256	8.76×10^{-3}	4.47×10^{-3}	49.7
Adult window	25,856	7.71×10^{-3}	7.60×10^{-3}	1.44
UNSW window	524,288	1.77×10^{-4}	1.76×10^{-4}	1.01

Table 5 makes the scale of these gains explicit. Large relative reductions on enriched blocks coexist with modest improvements on complete windows. The two protocols test whether better assignments can be found at their respective budgets, so their reductions should be interpreted within each setting.

Two further controls assess transfer of the observed effects. Predicted, constant and training-rate design references preserve positive median reductions when evaluated in the common label-derived world (Table 31); their information sources are specified in Appendix D.7. A nonlinear Adult scorer also preserves the assignment ordering across both shards and all capacities (Table 29). It supplies another fixed rank source within the theory's scorer-agnostic formulation.

7 Conclusion

In this paper, we asked which bag assignment leaves a fixed model's AUC least undetermined once group means are released. When rows carry separate response probabilities, the width is within-bag rank transport. For distinct scores and equal capacities with integer $n\pi$, adjacent-rank grouping minimises worst-case width at fixed prevalence and coincides with the partition that training-oriented curation selects. Repeated observable states instead tie one probability to several bag means, and joint endpoint programs with capacity-preserving swaps show that the two objectives then select different bags. The experiments connect these mechanisms to narrower evaluation intervals across click-log, census and network-traffic workloads, where direct width minimisation also beats the risk objective. Bag membership therefore determines how much a release reveals about a model's ranking performance.

The shared-state analysis assumes exact population means and reference probabilities specified before feedback. At full-window scale the search returns the narrowest assignment it reaches within a fixed candidate budget.

AI USE STATEMENT

In this work, we used generative AI tools for tasks with required disclosure. Generative AI tools helped develop the theoretical models and conceptual frameworks, helped formulate the mathematical claims, provided critical ingredients for proving those claims, and assisted in the writing of the proofs. The mathematical claims, the proofs, and every experimental result were checked and verified by the authors. We additionally used generative AI tools for tasks with recommended disclosure, namely to aid and polish the writing and to edit the paper for readability. We have reviewed all AI-assisted work, including the mathematical derivations and the text. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Shivani Agarwal, Thore Graepel, Ralf Herbrich, Sariel Har-Peled, and Dan Roth. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, 2005.

Sushant Agarwal, Yukti Makhija, Rishi Saket, and Aravindan Raghuveer. Aggregating data for optimal learning. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 1–30. PMLR, 2025. URL https://proceedings.mlr.press/v286/agarwal25a.html.

Lorne Applebaum, Travis Dick, Claudio Gentile, Haim Kaplan, and Tomer Koren. Optimal learning from label proportions with general loss functions. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Ehsan Mohammady Ardehaly and Aron Culotta. Co-training for demographic classification using deep learning from label proportions. In *Proceedings of the IEEE International Conference on Data Mining Workshops*, pp. 1017–1024, 2017. doi: 10.1109/ICDMW.2017.144.

Takanori Asanomi, Shinnosuke Matsuo, Daiki Suehiro, and Ryoma Bise. MixBag: Bag-level data augmentation for learning from label proportions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16524–16533, 2023. doi: 10.1109/ICCV51070.2023.01519.

Denis Baručić and Jan Kybic. Fast learning from label proportions with small bags. In *Proceedings of the IEEE International Conference on Image Processing*, pp. 3156–3160, 2022. doi: 10.1109/ICIP46576.2022.9897895.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. https://archive.ics.uci.edu/dataset/2/adult.

Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

Anand Brahmbhatt, Mohith Pokala, Rishi Saket, and Aravindan Raghuveer. LLP-Bench: A large scale tabular benchmark for learning from label proportions. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 4374–4381, 2024. doi: 10.1145/3627673.3680032.

Róbert Busa-Fekete, Heejin Choi, Travis Dick, Claudio Gentile, and Andres Munoz Medina. Easy learning from label proportions. In *Advances in Neural Information Processing Systems*, volume 36, pp. 14957–14968, 2023. doi: 10.52202/075280-0656.

Róbert István Busa-Fekete, Travis Dick, Claudio Gentile, Andrés Muñoz Medina, Adam Smith, and Marika Swanberg. Auditing privacy mechanisms via label inference attacks. In *Advances in Neural Information Processing Systems 37*, pp. 82823–82862, 2024. doi: 10.52202/079017-2634.

Robert Istvan Busa-Fekete, Travis Dick, Claudio Gentile, Haim Kaplan, Tomer Koren, and Uri Stemmer. Nearly optimal sample complexity for learning with label proportions. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5943–5976, 2025.

Lin Chen, Gang Fu, Amin Karbasi, and Vahab Mirrokni. Learning from aggregated data: Curated bags versus random bags. *arXiv preprint arXiv:2305.09557*, 2023.

Criteo AI Lab. Criteo 1tb click logs dataset. Online, 2015. URL https://ailab.criteo.com/download-criteo-1tb-click-logs-dataset/. Accessed: 2026-08-22.

Gabriel Dulac-Arnold, Neil Zeghidour, Marco Cuturi, Lucas Beyer, and Jean-Philippe Vert. Deep multi-class learning from label proportions. *arXiv preprint arXiv:1905.12909*, 2019. URL https://arxiv.org/abs/1905.12909.

Otis Dudley Duncan and Beverly Davis. An alternative to ecological correlation. *American Sociological Review*, 18(6):665–666, 1953. doi: 10.2307/2088122.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

Rodrigo Cabral Farias and Jean-Marc Brossier. Optimal scalar quantization for parameter estimation. *arXiv preprint arXiv:1310.6945*, 2013. URL https://arxiv.org/abs/1310.6945.

Noam Finkelstein, Roy Adams, Suchi Saria, and Ilya Shpitser. Partial identifiability in discrete data with measurement error. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1798–1808, 2021.

Benjamin Fish and Lev Reyzin. On the complexity of learning from label proportions. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 1675–1681, 2017. doi: 10.24963/ijcai.2017/232.

Walter D. Fisher. On grouping for maximum homogeneity. *Journal of the American Statistical Association*, 53(284):789–798, 1958. doi: 10.1080/01621459.1958.10501479.

Gabriel Franco, Giovanni Comarela, and Mark Crovella. Evaluating LLP methods: Challenges and approaches. *arXiv preprint arXiv:2310.19065*, 2023. URL https://arxiv.org/abs/2310.19065.

Allan Grønlund, Kasper Green Larsen, Alexander Mathiasen, Jesper Sindahl Nielsen, Stefan Schneider, and Mingzhou Song. Fast exact (k)-means, (k)-medians and Bregman divergence clustering in one dimension. *arXiv preprint arXiv:1701.07204*, 2017. URL https://arxiv.org/abs/1701.07204.

James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic curve. *Radiology*, 143(1):29–36, 1982. doi: 10.1148/radiology.143.1.7063747.

Shreyas Havaldar, Navodita Sharma, Shubhi Sareen, Karthikeyan Shanmugam, and Aravindan Raghuveer. Learning from label proportions: Bootstrapping supervised learners via belief propagation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KQe9tHd0k8.

Adel Javanmard, Lin Chen, Vahab Mirrokni, Ashwinkumar Badanidiyuru, and Gang Fu. Learning from aggregate responses: Instance level versus bag level loss functions. In *International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=E60SIDItyT.

Adel Javanmard, Matthew Fahrbach, and Vahab Mirrokni. Priorboost: An adaptive algorithm for learning from aggregate responses. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21410–21429, 2024b.

Wenxin Jiang, Gary King, Allen Schmaltz, and Martin A. Tanner. Ecological regression with partial identification. *Political Analysis*, 28(1):65–86, 2020. doi: 10.1017/pan.2019.19.

Shunsuke Kubo, Shinnosuke Matsuo, Daiki Suehiro, Kazuhiro Terada, Hiroaki Ito, Akihiko Yoshizawa, and Ryoma Bise. Theoretical proportion label perturbation for learning from label proportions in large bags. In *Proceedings of the 27th European Conference on Artificial Intelligence*, pp. 2194–2201, 2024. doi: 10.3233/FAIA240740.

Gene Li, Lin Chen, Adel Javanmard, and Vahab Mirrokni. Optimistic rates for learning from label proportions. In *Proceedings of the 37th Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 3437–3474, 2024.

Tianhao Ma, Ximing Li, Changchun Li, and Renchu Guan. Learning from label proportion with dual-proportion constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 7583–7592, 2026.

Yukti Makhija and Rishi Saket. Weak to strong learning from aggregate labels. In *Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 2876–2891, 2025.

Henry B. Mann and Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947.

Francesca Molinari. Microeconometrics with partial identification. *arXiv preprint arXiv:2004.11751*, 2020.

Sarah Moon. Partial identification of individual-level parameters using aggregate data in a nonparametric model. *Econometric Reviews*, 45(5):681–702, 2026. doi: 10.1080/07474938.2025.2604682.

Nour Moustafa and Jill Slay. UNSW-NB15: A comprehensive data set for network intrusion detection systems. In *Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, 2015.

David R. Musicant, Janara M. Christensen, and Jamie F. Olson. Supervised learning by training on aggregate outputs. In *Proceedings of the Seventh IEEE International Conference on Data Mining*, pp. 252–261, 2007. doi: 10.1109/ICDM.2007.50.

Ron Nafshi and Maggie Makar. Partial identification of the maximum mean discrepancy with mismeasured data. In *Proceedings of the 40th Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pp. 2623–2645, 2024.

Frank Nielsen and Richard Nock. Optimal interval clustering: Application to Bregman clustering and statistical mixture learning. *IEEE Signal Processing Letters*, 21(10):1289–1292, 2014. doi: 10.1109/LSP.2014.2333001.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Felipe Maia Polo, Subha Maity, Mikhail Yurochkin, Moulinath Banerjee, and Yuekai Sun. Weak supervision performance evaluation via partial identification. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4309.

Novi Quadrianto, Alex J. Smola, Tibério S. Caetano, and Quoc V. Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10:2349–2374, 2009.

Rishi Saket. Algorithms and hardness for learning linear thresholds from label proportions. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1267–1279, 2022. doi: 10.52202/068431-0093.

Rishi Saket, Aravindan Raghuveer, and Balaraman Ravindran. On combining bags to better learn from label proportions. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 5913–5927, 2022.

Jonas Schweisthal, Dennis Frauen, Maresa Schröder, Konstantin Hess, Niki Kilbertus, and Stefan Feuerriegel. Learning representations of instruments for partial identification of treatment effects. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53703–53727, 2025.

Clayton Scott and Jianxin Zhang. Learning from label proportions: A mutual contamination framework. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22256–22267, 2020.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, et al. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Haizhou Wang and Mingzhou Song. Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming. *The R Journal*, 3(2):29–33, 2011. doi: 10.32614/RJ-2011-015.

Zixi Wei, Lei Feng, Bo Han, Tongliang Liu, Gang Niu, Xiaofeng Zhu, and Heng Tao Shen. A universal unbiased method for classification from aggregate observations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36804–36820, 2023.

Zheng Xie, Yu Liu, Hao-Yuan He, Ming Li, and Zhi-Hua Zhou. Weakly supervised AUC optimization: A unified partial AUC approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. doi: 10.1109/TPAMI.2024.3357814.

Felix X. Yu, Dong Liu, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. $\propto$SVM for learning with label proportions. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 504–512, 2013.

Felix X. Yu, Krzysztof Choromanski, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. On learning from label proportions. *arXiv preprint arXiv:1402.5902*, 2014. URL `https://arxiv.org/abs/1402.5902`.

Yivan Zhang, Nontawat Charoenphakdee, Zhenguo Wu, and Masashi Sugiyama. Learning from aggregate observations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7993–8005, 2020.

A Proofs and Program Details

This appendix proves the width results in Section 4 and supplies the endpoint and search details for Section 5.

A.1 Proof of Proposition 1

$$G_i = \frac{1}{N}\sum_{k=1}^{N}\left[\mathbb{1}\{s_i > s_k\} - \mathbb{1}\{s_i < s_k\}\right], \tag{3}$$

This subsection derives the affine AUC representation used in Section 4. **Statement of Proposition 1.** At fixed prevalence, the target in (1) satisfies $\mathrm{AUC}(s,\rho) = \frac{1}{2} + \left(\sum_{i=1}^{N}\rho_i G_i\right) / \left(2N\pi(1-\pi)\right)$.

Proof.

Write $h_{ik} = h(s_i, s_k)$ and use $h_{ik} + h_{ki} = 1$. The quadratic sum is $\sum_{i,k}\rho_i\rho_k h_{ik} = \frac{1}{2}N^2\pi^2$. The linear sum is $\sum_{i,k}\rho_i h_{ik} = \frac{1}{2}N^2\pi + \frac{N}{2}\sum_i \rho_i G_i$. Subtracting the quadratic sum gives the numerator $\frac{1}{2}N^2\pi(1-\pi) + \frac{N}{2}\sum_i \rho_i G_i$. Division by $N^2\pi(1-\pi)$ proves Proposition 1. The half-tie convention gives $h_{ik} + h_{ki} = 1$ also for tied scores.

A.2 Proof of Proposition 6

For a stochastic aggregate observation Y_B, write $P_B(\cdot \mid \rho)$ for its conditional law. Given a release y, define $\mathcal{I}_Y(B, y) = \{\widetilde{\rho} \in \mathcal{R}_\pi : y \in \operatorname{supp} P_B(\cdot \mid \widetilde{\rho})\}$. The AUC extrema over this set, when attained, are L_Y and U_Y. For binary count channels below, $w_i \mid \rho$ are independent Bernoulli outcomes, called witnesses. If a release depends on the world only through the population bag means, worlds sharing those means have the same release law. Their exact-mean compatible set is therefore contained in the support-based set for a realised release in that law's support. The resulting width comparison is an identification statement, with no sampling-confidence assertion.

Call a channel *witness blind* when its conditional observation law is independent of the latent world: $P_B(y \mid \rho) = P_B(y)$ for every $\rho \in \mathcal{R}_\pi$. The following proposition states what this independence implies for AUC evaluation.

Proposition 6 (Assignment disappears from witness-blind feedback)**.** *For a witness-blind channel and any release of positive probability, $\mathcal{I}_Y(B, y) = \mathcal{R}_\pi$ for every feasible partition B. All partitions therefore have the same sharp AUC width.*

The release leaves every fixed-prevalence world compatible, so changing the bags leaves the AUC range unchanged. The proof below establishes this identity. This establishes that informative bag design requires a channel whose law depends on the latent world.

Proof.

Witness blindness means $P_B(y \mid \rho) = P_B(y)$ for every ρ. For any release of positive probability, every world in $\mathcal{R}_\pi$ is therefore compatible, so the compatible set equals $\mathcal{R}_\pi$ for every feasible B. Sharp width is the range of the same target over this common set, hence it does not depend on the assignment.

A.3 Proof of Proposition 7

For conditionally independent binary bag releases based on positive counts, a nonconstant count-response function permits design-dependent observation laws for some world at each fixed prevalence when $m \geq 2$. The construction below proves this condition by exchanging rows between labelled bags while keeping their capacities fixed. Thus informative channels can make the choice of bag membership consequential.

This subsection supports the count-channel condition in Section 4.1. To locate this dependence, consider a binary release Y_j based on the positive count $C_j = \sum_{i \in B_j} w_i$ in bag j. An exchangeable count channel uses only this count, through $\mathbb{P}(Y_j = 1 \mid w_{B_j}) = \psi(C_j)$. Here w_{B_j} is the vector of outcomes in bag j and $\psi : \{0, \ldots, n\} \to [0, 1]$ specifies the release probability. We take the bag releases to be conditionally independent given the outcomes. To separate the contributions of different count orders, expand $\psi(C_j)$ as $\sum_{k=0}^{n} \widehat{\psi}_k \binom{C_j}{k}$. The coefficient $\widehat{\psi}_k$ is the kth forward difference of ψ at zero, obtained by repeatedly subtracting adjacent function values. Taking expectations yields the assignment-dependent release probability:

$$\mathbb{P}(Y_j = 1) = \sum_{k=0}^{n} \widehat{\psi}_k \, e_k(\rho_{B_j}), \tag{4}$$

where ρ_{B_j} is the bag's probability vector and $e_k(\rho_{B_j}) = \sum_{S \subseteq B_j, \, |S|=k} \prod_{i \in S} \rho_i$. Thus e_k sums probability products over all k-row subsets, with $e_0 = 1$. The identity follows because $\binom{C_j}{k}$ counts the subsets whose outcomes are all positive. The $k = 0$ term is constant, whereas each higher-order term uses the rows placed in that bag. We can now state when these terms permit different assignments to produce different feedback laws.

Proposition 7 (Active witness moments permit design dependence)**.** *Consider the conditionally independent count channel above with $m \geq 2$. If $\widehat{\psi}_k = 0$ for every $k \geq 1$, the observation law is assignment invariant. If at least one non-constant coefficient is active, then for every $\pi \in (0, 1)$ there exist a world $\rho \in \mathcal{R}_\pi$ and two feasible labelled assignments with different observation laws.*

An active coefficient means that some $\widehat{\psi}_k \neq 0$ with $k \geq 1$. Assignment can then change the subsets whose probability products enter the release law. Labelled assignments retain the bag indices, so the comparison concerns the joint release vector in that fixed order. The proof below constructs a swap that changes a labelled bag's release probability. The condition $m \geq 2$ supplies the two bags needed for this exchange.

Proof.

Every function on $\{0, \ldots, n\}$ has the finite Newton representation $\psi(c) = \sum_{k=0}^{n} \widehat{\psi}_k \binom{c}{k}$. Here $\widehat{\psi}_k$ is the kth forward difference of ψ at zero. For independent Bernoulli variables, $\binom{C_j}{k}$ counts the k-subsets of successful witnesses, so

$$\mathbb{E}\binom{C_j}{k} = \sum_{S \subseteq B_j, \, |S|=k} \prod_{i \in S} \rho_i = e_k(\rho_{B_j}), \tag{5}$$

and linearity yields (4). When every $\widehat{\psi}_k$ with $k \geq 1$ vanishes, ψ is constant. Every bag observation then has the same response probability for all B and ρ. Conditional independence across bags makes the joint law assignment invariant.

For the converse, define

$$F(x_1, \ldots, x_n) = \mathbb{E}[\psi(\textstyle\sum_{i=1}^{n} W_i)], \qquad W_i \sim \text{Bernoulli}(x_i) \tag{6}$$

with independent W_i. If some $\widehat{\psi}_k$ with $k \geq 1$ is active then ψ is non-constant, and evaluating F at a binary vector with c ones gives $F = \psi(c)$, so F is a non-constant polynomial. When $n = 1$, choose probabilities $\pi - \delta$ and $\pi + \delta$ in two bags for a sufficiently small $\delta > 0$. Swapping these rows changes the response probability of each labelled bag.

Now let $n \geq 2$. A non-constant polynomial cannot be constant on an open neighbourhood of $\pi\mathbf{1}$ for $0 < \pi < 1$. Hence there are $q \in (0,1)^{n-1}$ and distinct $a, b \in (0,1)$, all arbitrarily close to π, with $F(q, a) \neq F(q, b)$. Place the coordinates (q, a) in one labelled bag and b in a second bag, and choose the remaining probabilities near π so that their total is $N\pi$. Swapping the rows with probabilities a and b preserves the world and every capacity while changing the response probability of the bag containing (q, a) from $F(q, a)$ to $F(q, b)$, so the two assignments have different observation laws.

A.4 Proof of Theorem 2

$$\mathsf{W}(B \mid b) = \frac{1}{2N\pi(1-\pi)} \sum_{j=1}^{m} \Delta(G_{B_j}, b_j), \tag{7}$$

$$\mathcal{I}_{\text{mean}}(B, b^{\text{rel}}) = \{\widetilde{\rho} \in \mathcal{R}_\pi : \textstyle\sum_{i \in B_j} \widetilde{\rho}_i = b_j^{\text{rel}},\ j = 1, \ldots, m\}, \tag{8}$$

This subsection proves the bagwise width decomposition in Section 4. **Statement of Theorem 2.** Fix the AUC target and prevalence under exact-mean feedback. Suppose every row has its own probability coordinate. For any balanced partition B and feasible budget vector b, the sharp width is

$$\mathsf{W}(B \mid b) = \frac{1}{2N\pi(1-\pi)} \sum_{j=1}^{m} \Delta(G_{B_j}, b_j),$$

where G_{B_j} collects the rank contrasts of rows in bag j. Both endpoints are attained by bagwise continuous-knapsack solutions.

Proof.

The feasible set in (2) factors as

$$\mathcal{I}(B) = \prod_{j=1}^{m} \left\{ v \in [0,1]^{B_j} : \mathbf{1}^\top v = b_j \right\}. \tag{9}$$

The denominator in Proposition 1 is fixed because $\sum_j b_j = N\pi$, and its variable numerator is a sum of bagwise linear objectives. A global maximum is therefore obtained by maximising each bag objective and a global minimum by minimising each one.

For one bag the continuous-knapsack exchange applies. If $g_p > g_q$ with $v_p < 1$ and $v_q > 0$, shifting a positive amount from v_q to v_p raises the objective and preserves the budget. Repeating the exchange fills the coordinates in decreasing order of coefficient value. With $b = k + f$, the maximising vector assigns one to the leading k ordered coordinates. It assigns f to coordinate $k + 1$ and zero to the remaining coordinates. The minimising vector reverses this coefficient order. Their difference is (17). Summing the bag ranges and applying the AUC normalisation proves (7). The constructed vectors attain both endpoints, so the range is sharp.

A.5 Proof of Proposition 3

When $n\pi$ is an integer, the expected random width equals $n(N+1)/\{N(n+1)\}$.

$$\begin{aligned}
\mathsf{W}(B^{\mathrm{sc}} \mid \pi\mathbf{1}) &= \frac{m\Psi(n\pi, n)}{N^2\pi(1-\pi)}, \\
\mathsf{W}(B^{\mathrm{rr}} \mid \pi\mathbf{1}) &= m\,\mathsf{W}(B^{\mathrm{sc}} \mid \pi\mathbf{1}), \\
\mathbb{E}_B\mathsf{W}(B \mid \pi\mathbf{1}) &= \frac{N+1}{n+1}\mathsf{W}(B^{\mathrm{sc}} \mid \pi\mathbf{1}),
\end{aligned} \tag{10}$$

This subsection derives the canonical design comparisons in Section 4. **Statement of Proposition 3.** In the canonical design class at the flat reference world, the three widths are

$$\begin{aligned}
\mathsf{W}(B^{\mathrm{sc}} \mid \pi\mathbf{1}) &= \frac{m\Psi(n\pi, n)}{N^2\pi(1-\pi)}, \\
\mathsf{W}(B^{\mathrm{rr}} \mid \pi\mathbf{1}) &= m\,\mathsf{W}(B^{\mathrm{sc}} \mid \pi\mathbf{1}), \\
\mathbb{E}_B\mathsf{W}(B \mid \pi\mathbf{1}) &= \frac{N+1}{n+1}\mathsf{W}(B^{\mathrm{sc}} \mid \pi\mathbf{1}),
\end{aligned}$$

where the expectation is over uniform balanced partitions. If $n\pi$ is an integer, the contiguous width reduces to $1/m$ and the round-robin width to 1. The expected random width then equals $n(N+1)/\{N(n+1)\}$.

Proof.

To obtain the spacing formula, suppose that $g_1 > \cdots > g_n$ is an arithmetic progression with adjacent difference d. Substitution in (17) gives

$$\Delta(g, b) = d\,[b(n-b) - f(1-f)] = d\,\Psi(b, n). \tag{11}$$

For distinct global scores, adjacent normalised rank contrasts differ by $2/N$, and every score-contiguous bag is an arithmetic progression with this difference. With $b = n\pi$, Theorem 2 gives $\mathsf{W}(B^{\mathrm{sc}} \mid \pi\mathbf{1}) = m(2/N)\Psi(n\pi, n)/\{2N\pi(1-\pi)\}$, which is the score-contiguous identity in (10).

A round-robin bag contains ranks $j, j+m, \ldots, j+(n-1)m$, so its contrast difference is $2m/N$ and (11) multiplies every bag range by m, which is the second identity.

For a uniform random balanced partition, the marginal law of one bag is a uniform n-subset of $\{1, \ldots, N\}$. If its ordered ranks are $R_{(1)} < \cdots < R_{(n)}$, symmetry of discrete order statistics gives $\mathbb{E}R_{(t)} = t(N+1)/(n+1)$. The range in (17) is a fixed linear combination of ordered contrasts, and G is affine in rank, so expectation passes through the combination. The expected adjacent contrast difference is $2(N+1)/\{N(n+1)\}$, which is $(N+1)/(n+1)$ times the contiguous difference $2/N$. Summing over bags gives the third identity. If $b = n\pi$ is integral then $f = 0$ and $\Psi(b, n) = b(n-b) = n^2\pi(1-\pi)$, and substituting $N = mn$ yields the reduced values $1/m$, 1 and $n(N+1)/\{N(n+1)\}$.

A.6 Flat-World Lower Bound

This subsection proves the flat-world claim of Theorem 4. **Statement of Theorem 4.** In the canonical design class, score-contiguous assignment minimises width at the flat world. If in addition $n\pi$ is an integer, it solves the minimax problem in (13) with value $1/m$. The flat world then attains the largest width for B^{sc}.

Proof.

Write the ordered global contrasts as $g_{(1)} \le \cdots \le g_{(N)}$. Between two adjacent contrast values a threshold splits the ordered pool after i elements. Let $\alpha_j \in \{0, \ldots, n\}$ be the number of below-threshold elements in bag j, so that $\sum_j \alpha_j = i$. The substitution $v \mapsto \mathbf{1} - v$ maps the feasible set at budget b onto the feasible set at budget $n-b$ and shifts $g^\top v$ by the constant $g^\top\mathbf{1}$, so $\Delta(g, b) = \Delta(g, n-b)$ and it suffices to treat the reduced budget $b' = \min(b, n-b)$. Writing $b' = k' + f'$ with $k' = \lfloor b' \rfloor$, define the threshold cost

$$\zeta(d) = \min(k', d) + f'\mathbb{1}\{d > k'\} = \min(b', d), \qquad d \in \{0, \ldots, n\}. \tag{12}$$

The within-bag range has a layer representation whose contribution at this threshold is $\zeta(\min(\alpha_j, n - \alpha_j))$, because filling the top coordinates attains every threshold maximum at once and filling the bottom coordinates attains every threshold minimum. The map ζ is non-decreasing and subadditive

on non-negative integers, and $\sum_j \min(\alpha_j, n-\alpha_j) \geq \min(i \bmod n,\, n-(i \bmod n))$. Applying subadditivity produces a pointwise lower bound at each threshold. A score-contiguous partition attains it because only one bag can be split at a threshold between adjacent global ranks. Integrating the threshold contributions proves that score-contiguous assignment minimises flat-world width. The quantity Ψ in (11) is itself invariant under $b \leftrightarrow n-b$, so the reduction to b' leaves the widths stated in Proposition 3 and Theorem 4 unchanged.

Complete enumeration of the balanced partitions reproduces this conclusion at 102 budget instances spanning $4{,}177{,}243$ distinct partitions. The instances take every capacity in $\{3,4,5,6\}$ together with every pool size up to sixteen rows that the capacity divides. Every budget lies strictly above $n/2$ in quarter steps, so each one is a budget that the reduction to b' actually moves, and 83 of them are not integers. The score-contiguous partition attains the minimum width in each instance and matches $m\Psi(b,n)/\{N^2\pi(1-\pi)\}$ to within 7×10^{-16}.

A.7 Minimax Part of Theorem 4

This subsection proves the minimax claim of Theorem 4. The two design objectives of Section 4.1 are

$$\min_{B \in \mathcal{B}_{m,n}} \mathsf{W}\big(B \mid b^{\mathrm{ref}}(B)\big) \qquad \text{and} \qquad \min_{B \in \mathcal{B}_{m,n}} \sup_{\rho^0 \in \mathcal{R}_\pi} \mathsf{W}(B \mid \rho^0), \tag{13}$$

where the first objective is pointwise at a specified reference world and the second minimises the largest width over the fixed-prevalence class.

Statement of Theorem 4. In the canonical design class, score-contiguous assignment minimises width at the flat world. If in addition $n\pi$ is an integer, it solves the minimax problem in (13) with value $1/m$. The flat world then attains the largest width for B^{sc}.

Proof.

For every design B,

$$\sup_{\rho^0 \in \mathcal{R}_\pi} \mathsf{W}(B \mid \rho^0) \geq \mathsf{W}(B \mid \pi\mathbf{1}) \geq \frac{1}{m}. \tag{14}$$

The second inequality combines the flat-world minimality established in Appendix A.6 with the contiguous value of Proposition 3 at integral $n\pi$. It remains to show that B^{sc} meets this bound. Every contiguous bag has contrast spacing $2/N$, so its width contribution is proportional to $\Psi(b_j, n)$. On each interval $b \in [k, k+1]$ direct expansion gives $\Psi(b,n) = b(n-2k-1) + k(k+1)$. The slopes $n-1, n-3, \ldots, 1-n$ decrease with k, so Ψ is concave, and under $\sum_j b_j = N\pi$ Jensen's inequality gives $\sum_j \Psi(b_j, n) \leq m\Psi(n\pi, n)$. When $n\pi$ is integral the normalised right side equals $1/m$. The flat world realises equal budgets and attains the bound, which with (14) completes the proof.

For every $n\pi$ in the same distinct-score class, $\sup_{\rho^0 \in \mathcal{R}_\pi} \mathsf{W}(B^{\mathrm{sc}} \mid \rho^0) = m\Psi(n\pi, n)/\{N^2\pi(1-\pi)\} \leq 1/m$. The concavity argument above gives this bound without an integer-budget restriction.

For noninteger budgets, complete enumeration on the instances in Table 6 also finds the contiguous partition to attain the minimum worst-case width. Each supremum is optimised over the continuous fixed-prevalence world class.

Table 6: Noninteger-budget instances. The minimax width equals both the contiguous supremum and the general closed form in every listed instance.

Rows	Capacity	Prevalence	Minimax width	Contiguous supremum	Closed form
6	3	0.17	0.4016064257	0.4016064257	0.4016064257
6	3	0.31	0.4830917874	0.4830917874	0.4830917874
6	3	0.43	0.4533297067	0.4533297067	0.4533297067
8	4	0.17	0.4518072289	0.4518072289	0.4518072289
8	4	0.31	0.4733520337	0.4733520337	0.4733520337
8	4	0.43	0.4742962056	0.4742962056	0.4742962056
12	6	0.17	0.4980707142	0.4980707142	0.4980707142
12	6	0.31	0.4921822243	0.4921822243	0.4921822243
12	6	0.43	0.4861961104	0.4861961104	0.4861961104

A.8 Proof of Proposition 8

In the separate-row setting, width is zero exactly when every bag has a boundary budget or constant rank contrasts. Tied-score bags can therefore identify AUC without identifying each probability.

This subsection supports the target-identification condition in Section 4. A final question is when the width vanishes entirely. Equation (7) reduces this question to the probability movements permitted within each bag.

Proposition 8 (Target identification without world identification)**.** *In the separate-row exact-mean setting,* (7) *equals zero if and only if every bag has $b_j \in \{0, n\}$ or constant G_i within that bag.*

A boundary budget fixes every probability in its bag. Constant contrasts instead let probability move without changing the target. For example, a bag containing only tied scores can leave individual probabilities unresolved while fixing its contribution to AUC. The proof below examines the nonnegative bag ranges.

Proof.

Every term in (7) is non-negative. For $b = 0$ or $b = n$ the capped simplex contains one point and $\Delta(g, b) = 0$. For $0 < b < n$ the feasible set permits a positive mass exchange between any two coordinates that are not simultaneously fixed at a boundary. If two contrast values differ, choosing the exchange direction changes the linear objective, so its range is positive. A non-degenerate bag therefore has zero range exactly when every contrast in that bag is equal, and applying the argument to each term proves the claim.

A.9 Endpoint Programs and Their Duals

This subsection supports the coupled endpoint construction in Section 5.1. **Statement of Proposition 5.** Fix the AUC target and prevalence under exact-mean feedback. For an equal-capacity assignment B and any feasible budget b, define the upper and lower numerator endpoints by

$$V^{\pm}(B \mid b) = \begin{Bmatrix} \max \\ \min \end{Bmatrix} \{c^\top \widetilde{u} : A_B \widetilde{u} = b,\ 0 \le \widetilde{u} \le 1\}, \tag{15}$$

where V^+ uses maximisation and V^- uses minimisation. Then $L(B \mid b) = \frac{1}{2} + V^-(B \mid b)/\{2N\pi(1-\pi)\}$. The maximum $U(B \mid b)$ uses V^+ in the same expression, and $\mathsf{W}(B \mid b) = \{V^+(B \mid b) - V^-(B \mid b)\}/\{2N\pi(1-\pi)\}$. Both programs attain their optima on the nonempty closed and bounded feasible set.

Proof.

For a fixed shared-atom assignment and a feasible budget b, the upper endpoint program is

$$\max_{\widetilde{u} \in \mathbb{R}^K} c^\top \widetilde{u} \quad \text{subject to} \quad A_B \widetilde{u} = b,\ 0 \le \widetilde{u}_a \le 1,\ a = 1, \dots, K, \tag{16}$$

and the lower program replaces the maximisation by minimisation. Both return extrema of the affine numerator, which Proposition 5 maps to the AUC endpoints. For pointwise comparison $b = A_B u^0$, so the fixed reference vector is feasible for every candidate. For post-design evaluation $b = b^{\mathrm{rel}}(B)$ and feasibility follows from the release itself. Overall prevalence is constant because every assignment partitions the same row multiset.

The equality multipliers give each bag budget a shadow value, and the bound multipliers account for atom probabilities that reach zero or one at an endpoint. A wide interval arises when c has a large component along feasible directions in the null space of A_B. Assignment changes the incidence columns and therefore changes which target directions the released equalities constrain. If c lies in the row span of A_B then $c^\top \widetilde{u}$ is fixed by $A_B \widetilde{u} = b$ and the width is zero for every feasible b, which recovers the constant-contrast branch of Proposition 8. In the separate-row model that row span consists of vectors constant within bags. Saturated budgets give the other branch, because their feasible coordinates are fixed at the box boundary. Shared atoms let row combinations constrain more general target directions.

For pointwise design, let $u^0 \in [0, 1]^K$ be the reference atom probabilities specified before feedback, with $N^{-1} \sum_i u^0_{a(i)} = \pi$. The reference budget is $b^{\mathrm{ref}}(B) = A_B u^0$. The same u^0 is feasible for every

candidate, so each candidate has well-defined endpoints at the common prevalence. After feedback, substituting $b = b^{\mathrm{rel}}$ gives the evaluation interval for the selected assignment.

A.10 The Twelve-Row Objective-Gap Instance

This subsection supplies the explicit risk–width separation used in Section 5.1.

The instance quoted in Section 5 takes $s = (1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6)$ with atom labels $(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6)$ and reference world $u^0 = (0.08, 0.16, 0.26, 0.38, 0.52, 0.66)$, split into three bags of capacity four. Complete enumeration of the 5,775 unlabelled balanced partitions returns one risk minimiser and eight width minimisers whose objective sets do not intersect. The risk-optimal partition has sharp width 0.2095 and the width-optimal partitions have 0.03573, a ratio of 5.86.

Why prediction risk and identification width can differ. The incidence matrix controls joint ambiguity, whereas prediction-based grouping controls the variation of scores inside each bag. To compare these objectives, let $\overline{s}_j = n^{-1} \sum_{i \in B_j} s_i$ be the bag mean of the fixed scorer outputs. The within-bag squared criterion is $R(B) = \sum_j \sum_{i \in B_j} (s_i - \overline{s}_j)^2$. This is the one-dimensional partition objective used by PriorBoost (Javanmard et al., 2024b). We apply it to fixed scorer outputs, keeping the model unchanged during design. Minimising R groups similar score values, while minimising width restricts the attainable rank-weighted sum over all compatible atom worlds.

A.11 Deterministic Search Rules

Algorithm 1 implements the pointwise objective in Section 5.2. Its reference vector is fixed before all starts.

Algorithm 1 Multi-start steepest sharp-width descent

Require: scores, atom map, u^0, capacities, primary partition, R random starts, $\varepsilon_{\mathrm{cmp}}$, $T_{\max}$
1: form distinct balanced starts, including the primary partition
2: $B_{\mathrm{best}} \leftarrow \varnothing$, $w_{\mathrm{best}} \leftarrow +\infty$
3: **for all** starting partitions $B^{(0)}$ **do**
4: $B \leftarrow B^{(0)}$, $t \leftarrow 0$, status $\leftarrow$ step limit
5: evaluate $w \leftarrow \mathsf{W}(B \mid A_B u^0)$ with both endpoint LPs
6: **while** $t < T_{\max}$ **do**
7: evaluate all cross-bag swaps in stable lexicographic order
8: select the steepest decrease exceeding $\varepsilon_{\mathrm{cmp}}$, resolving numerical ties by traversal order
9: **if** no eligible swap exists **then**
10: status $\leftarrow$ local stop; **break**
11: **end if**
12: update B, w to the selected candidate; $t \leftarrow t + 1$
13: **end while**
14: store this trajectory's $B, w, t,$ status
15: update $B_{\mathrm{best}}, w_{\mathrm{best}}$ if w is smaller beyond tolerance, or tied with a smaller canonical partition key
16: **end for**
17: **return** $B_{\mathrm{best}}, w_{\mathrm{best}}$ and the stored trajectory records

Traversal and numerical evaluation. Rows within bags are sorted by stable index. Bag pairs and then row pairs are visited lexicographically. A candidate replaces the incumbent when its width is smaller by more than $\varepsilon_{\mathrm{cmp}}$; numerical ties retain the earlier move. Across starts, ties use canonical unlabelled partition order. Random starts exclude duplicate unlabelled partitions and the primary start. The reported block search uses $R = 16$, $\varepsilon_{\mathrm{cmp}} = 10^{-12}$ and $T_{\max} = 100$. The step limit counts accepted swaps when every pass improves. A final pass with no eligible move records a local stop.

Endpoint evaluation uses SciPy's HiGHS interface (Virtanen et al., 2020). Both solves must report success. A failed solve raises an error in the reference-world search; the implementation does not assign it a substitute width or silently skip it. HiGHS uses its default feasibility settings. The 10^{-12} comparison tolerance is not an LP residual acceptance threshold. The supplementary evaluator records

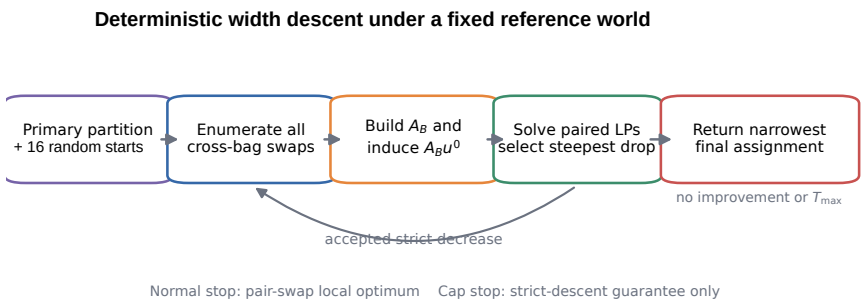

Figure 2: The width-directed search. A partition fixes the bag–atom incidence matrix, the reference world induces its probability masses, and two bounded LPs return the sharp endpoints. An accepted pair swap preserves capacity and decreases the computed width by more than $\varepsilon_{\mathrm{cmp}}$.

equality residuals and caches widths by the canonical bag–atom multisets. Its empirical-release comparisons record infeasible constraints separately (Appendix E.5).

Neighbourhood and cost. Pair swaps are the smallest membership changes preserving equal capacities. With $Q = \binom{m}{2}n^2$ candidate swaps per pass, a trajectory accepting T moves evaluates at most $1 + (T+1)Q$ widths, including its initial assignment and a possible final unsuccessful pass. Across $R+1$ starts, the upper bound is $\sum_{r=0}^{R}[1 + (T_r+1)Q]$. Each uncached width evaluation solves two LPs, each with K variables, m equality constraints and box bounds. This is a count of optimisation calls, not their running time. The original evaluator solves each call, while the supplementary cache avoids repeated LP pairs without changing the candidate-call budget. Sorting alone gives the proven canonical minimax solution under Theorem 4's assumptions. Restricted-budget variants are evaluated in Section 6.

A.12 Width Decomposition and Design Interpretation

This subsection expands the within-bag calculation and design interpretations in Section 4.

To find the maximum, we move probability towards larger coefficients until each reaches its upper bound. The minimum fills the smallest coefficients instead. This continuous-knapsack calculation is a linear optimisation with a single sum constraint (Bertsimas & Tsitsiklis, 1997). It fits the present constraints because moving mass within a bag preserves its released budget.

Order the coefficients as $g_{(1)} \geq \cdots \geq g_{(n)}$. Writing $k = \lfloor b \rfloor$ and $f = b - k$, the difference between the two endpoints is

$$\Delta(g, b) = \sum_{t=1}^{k} \left(g_{(t)} - g_{(n+1-t)}\right) + f\left(g_{(k+1)} - g_{(n-k)}\right) \mathbb{1}\{f > 0\}, \tag{17}$$

where k counts the fully assigned coordinates and $f \in [0, 1)$ is the remaining fractional mass. The summation pairs the k largest coefficients with the k smallest. The fractional term accounts for the remaining mass at each endpoint. When $f = 0$, that term is defined as zero without evaluating its indexed coefficients. In particular, budgets $b = 0$ and $b = n$ leave no within-bag freedom. Theorem 2 combines these single-bag ranges into the exact AUC width.

Each bag's endpoint assignment can be chosen independently because no probability coordinate appears in another bag. Combining these assignments attains the global endpoints, as proved in Appendix A.4. Pointwise design uses $b = b^{\mathrm{ref}}(B)$, while evaluation after feedback uses $b = b^{\mathrm{rel}}$. The scorer fixes G and the prevalence fixes the normaliser. Assignment changes which contrasts appear in each summand and which reference budget they receive. Thus grouping nearby ranks restricts how far probability can move while preserving a bag mean.

Three canonical designs. We now compare assignments that place nearby ranks in the same bag or spread them across bags. For this comparison, the canonical design class uses separate row probabilities and distinct scores under fixed-prevalence exact-mean feedback. Partitions have the equal capacities specified in Section 3. We also use the flat reference world $\rho_i^0 = \pi$, meaning that

every row has the same response probability. Every balanced partition then has budget $n\pi$ in every bag, which isolates the contribution of rank geometry.

Let B^{sc} denote score-contiguous assignment, which groups consecutive ascending ranks into bags. Let B^{rr} denote round-robin assignment, which sends consecutive ranks to successive bags and repeats this cycle. The third design is a uniformly sampled balanced partition. To express their widths compactly, define $\Psi(b, n) = b(n - b) - f(1 - f)$ with $f = b - \lfloor b \rfloor$ as above. For equally spaced contrasts, this is the single-bag range divided by the contrast spacing. The term $b(n - b)$ gives the integer-budget expression, and $f(1 - f)$ adjusts for fractional mass. Proposition 3 uses the common flat reference world because every bag budget equals $n\pi$.

B Experimental Details

This appendix specifies the workloads and comparisons in Section 6.

Table 7: Distinct evaluation regimes and search budgets. Small-block references use raw feature keys; full-window shared states use complete preprocessed feature vectors. All scorers are fixed before evaluation.

Study	Rows and reference	Assignment or search
Separate-row spectrum	Criteo 1,048,576; Adult 16,384 / 8,192; UNSW 524,288. Binary outcome reference.	Contiguous, round-robin, ten balanced random and ten independent-random-score orders.
Shared-state final grid	Criteo and Adult: ten blocks per size 12,16,24,32, five per shard. Smoothed evaluation rates.	Primary plus 16 random starts, exhaustive pair swaps. Capacity four.
Calibration / objective gap	Two disjoint Criteo sets, ten twelve-row blocks each, preceding the final grid.	Complete 5,775-partition enumeration. Adult objective comparison uses its final twelve-row grid.
Larger Adult blocks	Five disjoint shard-0 blocks per size 64,128,256; smoothed evaluation rates.	One contiguous start, sampled first improvement, 10,000 proposals or 100 accepted swaps.
Shared-state full windows	Adult 25,856 and UNSW 524,288, including singleton states; smoothed evaluation-pool rates.	34 structured proposals per capacity 32,64,128,256. Best found within that family.

B.1 Data and Fixed Scorers

This subsection supplies the data and scorer settings for Section 6.1.

The advertising workload uses two chronological day parts of the Criteo 1TB click logs (Criteo AI Lab, 2015), corresponding to 15 and 16 February 2015. Each impression carries a click indicator, 13 integer features and 26 categorical features, and we use the 13 numeric features for the scorer and for state identity. The census workload uses the two released files of the UCI Adult income table (Becker & Kohavi, 1996) as two shards. Its five numeric columns comprise age and education number alongside capital gain and capital loss. Hours per week completes this feature set. The network workload uses part one of the UNSW-NB15 capture (Moustafa & Slay, 2015) with 13 numeric columns.

In all three cases non-negative values receive a $\log(1 + x)$ transform, missing values take fit-split medians, and standardisation uses fit-split means and standard deviations. The evaluation split never enters these preprocessing estimates. Scorers are logistic models with $C = 1$, L-BFGS and a 250-iteration limit (Pedregosa et al., 2011), and each converges before the limit. Both Criteo scorers stop after 14 L-BFGS iterations. The additional Adult comparison with a nonlinear scorer uses a two-layer network of 64 and 32 units with ReLU activation, the Adam solver, $\alpha = 10^{-4}$ and a 500-iteration limit.

B.2 Coordinate Regimes and Design Timeline

This subsection explains the design inputs and probability models in Sections 3 and 6.1.

Table 8: Evaluation workloads. Coverage is the share of evaluation rows whose observable state occurs at least twice. Criteo and Adult supply the shared-atom study, and the UNSW-NB15 shard is used for the row-coordinate study at advertising scale.

	Criteo		Adult		UNSW-NB15
Quantity	shard 0	shard 1	shard 0	shard 1	part 1
Scorer-fit rows	262,144	131,072	6,512	3,256	131,072
Evaluation shard rows	1,048,576	524,288	26,049	13,025	524,288
Spectrum window rows	1,048,576	–	16,384	8,192	524,288
Repeated atoms	1,722	920	2,659	1,493	–
Coverage	0.78%	0.78%	73.85%	67.60%	–
Maximum multiplicity	1,567	784	131	67	–
Positive rate	0.0325	–	0.2407	0.2359	0.0120

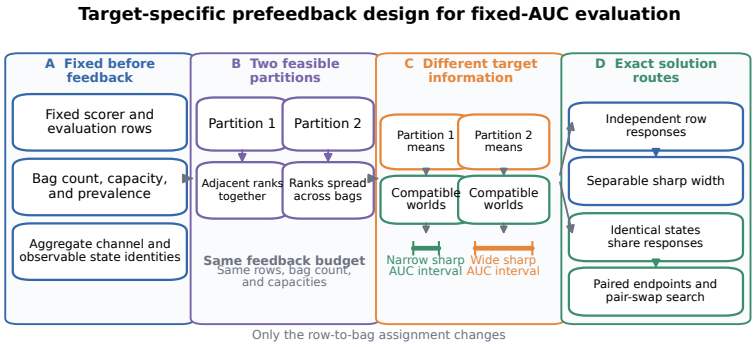

Figure 3: Pre-feedback partition design for fixed-AUC evaluation. The scorer and evaluation rows stay fixed. Capacities and prevalence also remain unchanged, as does the feedback channel. Two feasible partitions therefore differ only in incidence while inducing sharp AUC intervals of different width.

The row-coordinate study gives every evaluation row an separate latent coordinate even when observed numeric features coincide, and it uses the separable closed form of Theorem 2. The shared-atom study enforces one probability for every exact numeric state and uses the coupled programs of Proposition 5.

Both are pointwise comparisons at a specified reference world. We hold the scorer and rows fixed throughout each comparison. The atom map and capacities are also fixed. Each workload has one common reference world. The row-coordinate study uses the fixed evaluation outcome vector. The shared-atom study uses the Jeffreys-smoothed reference probabilities defined below. Assignment is the only intervened object, and for a candidate B the incidence matrix deterministically induces $b^{\mathrm{ref}}(B) = A_B u^0$, so candidate budgets vary with assignment even though the reference world is fixed.

B.3 Full-Shard Assignment Study

This subsection specifies the assignment comparisons in Section 6.2.

The Criteo study uses all $1{,}048{,}576$ evaluation rows of shard 0 in $4{,}096$ bags of capacity 256. The assignments are score-contiguous assignment, ten balanced random assignments and round-robin assignment. Ten independent-random-score assignments sort by independently generated random scores and then apply the same stable contiguous construction. The same 22 partitions are reused under the witness-blind channel. Every exact-mean budget is induced from the same fixed outcome vector. A feature-equal assignment based on the leading categorical feature supplies descriptive context only. It retains complete capacity-sized groups covering $561{,}152$ rows, or 53.52% of shard 0, and because its row set differs from the primary workload it is excluded from paired comparisons.

The new-domain spectra compare bag capacities $4, 8, 16, 32, 64, 128, 256$ on a power-of-two evaluation window of each shard, which is $16{,}384$ rows on Adult shard 0, $8{,}192$ rows on Adult shard 1 and $524{,}288$ rows on UNSW-NB15. The UNSW-NB15 scorer is fitted on $131{,}072$ disjoint rows so that capacity 128 reproduces the $4{,}096$ bags of the Criteo setting.

B.4 Exact-Repeat Blocks and Comparison Sets

This subsection defines the repeated-state blocks used in Section 6.4.

An atom is defined by exact equality of the selected raw numeric scorer-input keys (including their missing-value encoding), and atoms with two or more occurrences are eligible. A fixed label-independent pseudorandom order of feature keys determines atom allocation, and stable row order selects two copies per atom, so labels determine neither eligibility nor ordering nor block membership. For these label-independent blocks, all occurrences of atom a in its evaluation shard define the reference probability $u_a^0 = (y_a^+ + 0.5)/(M_a + 1)$, where M_a is the full-shard multiplicity and y_a^+ the positive count.

Table 9 separates the three comparison sets on Criteo. The leading 60 atoms in the fixed shard-0 order form the objective-gap set. The following 60 atoms form the method-calibration set. The final grid uses atoms after these 120, while shard 1 uses its own independent fixed order. The objective comparison, the search calibration and the final evaluation therefore do not reuse atoms. The Adult grid follows the same construction with 210 selected atoms per shard.

Table 9: Shared-atom evidence sets on Criteo. Every twelve-row design contains six atoms, three bags and capacity four.

Set	Shard	Blocks	Rows	Comparison purpose	Exact reference
Objective-gap	0	10	12	risk–width sets	5,775 partitions
Method-calibration	0	10	12	search calibration	5,775 partitions
Final grid	0 and 1	40	12, 16, 24, 32	scale and chronology	12 and 16 rows

The final grid keeps capacity four and contributes five blocks from each shard at each scale, with the coupled design size growing from 6 to 16 atoms as rows grow from 12 to 32. The Adult grid mirrors this layout exactly.

B.5 Comparators and Ablations

This subsection defines the reference assignments and search comparisons in Section 6.1.

The literature-derived comparator instantiates the one-dimensional within-bag squared-risk objective of PriorBoost (Javanmard et al., 2024b) on the fixed scorer outputs. It is a partition objective, distinct from the adaptive PriorBoost training system. On enumerable twelve-row blocks we recover all risk minimisers and use the one with the smallest sharp width, which is the conservative choice for the primary start. At 16 to 32 rows, stable score-contiguous grouping minimises the equal-capacity squared-score objective. Stable row order chooses among score ties. We call these scale-specific constructions the PriorBoost-style risk reference. Enumeration of all $2{,}627{,}625$ balanced sixteen-row partitions finds a unique risk minimiser in each of the twenty Criteo and Adult blocks. It is the score-contiguous partition and attains the narrowest risk-optimal width in each block.

Score-contiguous and round-robin assignments are theory-facing references. Balanced random assignment and the random-score construction give generic and independent-random-score comparisons. Algorithmic comparisons are the best unoptimised partition among the identical 17 starts and primary-only descent. Compute-matched random search evaluates the same number of unique random balanced partitions as the full method evaluates exact-width candidates, and it is defined only at 16 to 32 rows because every balanced twelve-row partition is enumerated.

B.6 Measures

This subsection specifies the width summaries used in Section 6.

Primary comparisons report the absolute sharp width. For full-shard random assignment t the normalised assignment effect is $E_t = \{\mathsf{W}(B_t^{\mathrm{random}} \mid \rho^0) - \mathsf{W}(B^{\mathrm{sc}} \mid \rho^0)\}/\mathsf{W}(B^{\mathrm{rr}} \mid \rho^0)$. For the objective-gap set, with $\mathcal{B}_R^\star$ and $\mathcal{B}_W^\star$ the complete risk- and width-minimiser sets, we report their intersection and the conservative normalised width regret

$$\mathrm{Reg}_W = \frac{\min_{B \in \mathcal{B}_R^\star} \mathsf{W}(B \mid \rho^0) - \min_B \mathsf{W}(B \mid \rho^0)}{\max_B \mathsf{W}(B \mid \rho^0) - \min_B \mathsf{W}(B \mid \rho^0)}, \tag{18}$$

which is defined when the width range is positive. On the final grid the relative reduction from the scale-specific risk reference is $D = \{\mathsf{W}(B^{\mathrm{risk}} \mid \rho^0) - \mathsf{W}(B^{\mathrm{method}} \mid \rho^0)\}/\mathsf{W}(B^{\mathrm{risk}} \mid \rho^0)$, defined for positive reference width. All quantities are paired at block level. Percentile bootstrap intervals use 10,000 fixed resamples (Efron & Tibshirani, 1993) and are 95% descriptive summaries of the observed block effects across the fixed shards. The 45 random-pair differences are likewise descriptive summaries of dependent comparisons.

C Complete Numerical Results

This appendix reports the advertising results and cross-workload comparisons underlying the assignment and search results in Section 6.

C.1 Full-Shard Assignment Variability

This subsection supports the assignment and feedback comparisons in Section 6.2.

Table 10 lists every random and independent-random-score width. The score-contiguous reference is 0.0002395 and the round-robin reference is 0.99614 in every row. The independent-random-score differences are small relative to the primary assignment effect. The independent-random-score-minus-random differences include both signs and their paired bootstrap mean interval contains zero. The older feature-equal subset in Figure 4 uses a different row set. The equal-row feature-bin comparison is reported separately in Table 21.

Table 10: Ten full-shard balanced random assignments on Criteo shard 0.

Rep.	Random	Random-score	Rep.	Random	Random-score
1	0.99248	0.99253	6	0.99226	0.99258
2	0.99242	0.99242	7	0.99238	0.99254
3	0.99238	0.99244	8	0.99233	0.99238
4	0.99242	0.99245	9	0.99239	0.99241
5	0.99230	0.99205	10	0.99241	0.99274

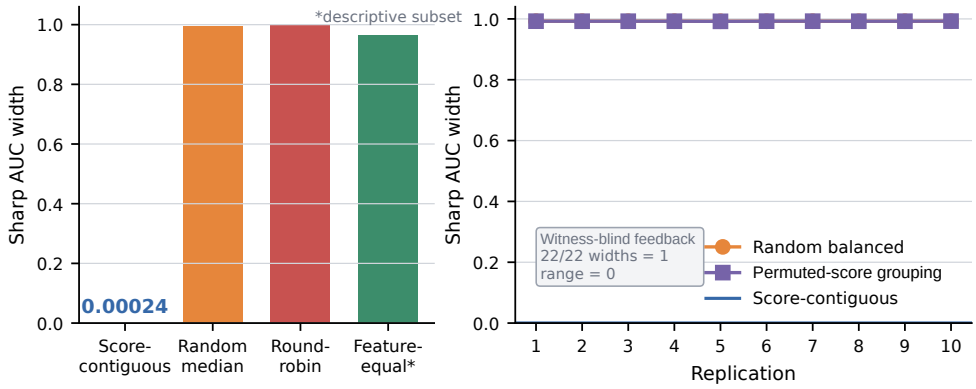

Figure 4: Assignment effect and mechanism comparisons on the full-shard row-coordinate workload. Panel (a) compares score-contiguous, balanced-random and round-robin assignment, with the feature-equal bar computed on a different descriptive subset. Panel (b) shows the ten paired random and independent-random-score widths, and its inset records the witness-blind comparison, for which all 22 assignments have width one.

C.2 Repeated-State Objective Gap

This subsection supports the risk–width comparison in Section 6.4.

Table 11 reports every normalised regret in the exact six-atom experiment, where each row evaluates all 5,775 balanced partitions and the risk- and width-minimiser sets never intersect.

Table 11: Objective-gap results by exact-repeat block.

Block	Regret	Intersection	Block	Regret	Intersection
1	0.175	0	6	0.1821	0
2	0.1684	0	7	0.1858	0
3	0.1786	0	8	0.1906	0
4	0.1852	0	9	0.1883	0
5	0.1686	0	10	0.1667	0

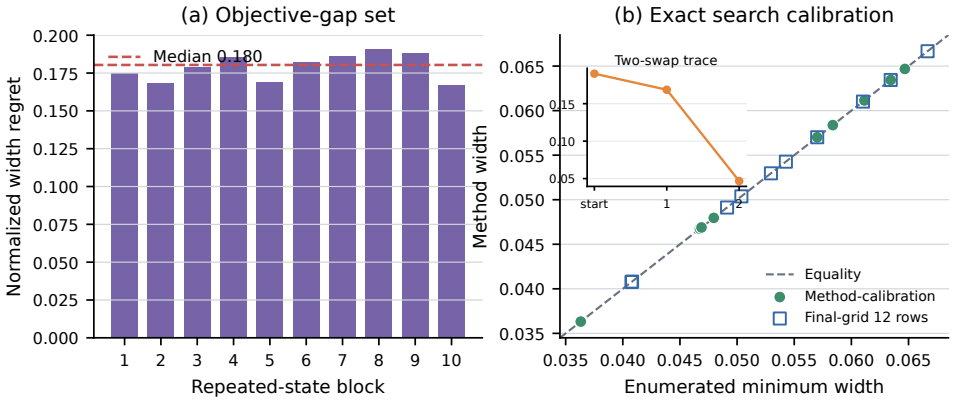

Figure 5: Objective separation and search calibration on Criteo. Panel (a) reports normalised width regret in the ten enumerated objective-gap blocks, where regret uses the narrowest partition in the complete risk-minimiser set. Panel (b) compares the returned width with the enumerated minimum on ten disjoint calibration blocks and ten further twelve-row blocks, with all 20 markers on the equality line. The inset follows two capacity-preserving swaps in one block.

C.3 Disjoint Method-Calibration Blocks

This subsection supports the exact-reference comparison in Section 6.4.

Table 12 reports the ten atom-disjoint comparison blocks evaluated with the same specified search. Every method width equals the minimum obtained by enumerating all 5,775 balanced partitions. The ten twelve-row blocks of the final grid also attain their enumerated minima, as the twelve-row rows of Tables 13 and 14 show.

Table 12: Exact search calibration on the disjoint method-calibration set.

Block	Risk	Method	Enumerated	Block	Risk	Method	Enumerated
1	0.19	0.04675	0.04675	6	0.2027	0.05701	0.05701
2	0.1974	0.06344	0.06344	7	0.1957	0.06114	0.06114
3	0.1986	0.03632	0.03632	8	0.1983	0.06468	0.06468
4	0.1918	0.05837	0.05837	9	0.2124	0.04796	0.04796
5	0.1897	0.05838	0.05838	10	0.2286	0.04689	0.04689

C.4 All Blocks of the Final Grid

This subsection gives the block-level results for the search comparison in Section 6.4.

Tables 13 and 14 give paired widths for all 20 blocks of each Criteo shard, rounded for display after comparisons. Numerically zero values are displayed as zero; comparisons use unrounded values.

C.5 Initialisation and Search Cost

This subsection supports the initialisation and computational-cost comparisons in Section 6.4.

Table 4 separates candidate calls, starts, accepted moves and stopping reasons. The full method improves on primary-only descent in 34 of 40 Criteo blocks. Its two wins and 38 ties against

Table 13: Complete paired results for Criteo shard 0.

Rows	Blk	Risk	Method	Reduction	Rows	Blk	Risk	Method	Reduction
12	1	0.1974	0.06344	0.6786	24	1	0.09832	0.02216	0.7746
12	2	0.1872	0.04081	0.782	24	2	0.1012	0.02174	0.7853
12	3	0.2174	0.04076	0.8125	24	3	0.1008	0.01815	0.82
12	4	0.2087	0.05038	0.7586	24	4	0.09825	0.02568	0.7386
12	5	0.2027	0.05701	0.7188	24	5	0.1037	0	1.000
16	1	0.1508	0.05273	0.6505	32	1	0.08054	0.01311	0.8372
16	2	0.1525	0.05647	0.6298	32	2	0.07128	0.01326	0.814
16	3	0.1476	0.06727	0.5442	32	3	0.07425	0.009166	0.8765
16	4	0.1447	0.06149	0.5749	32	4	0.07308	0.0184	0.7482
16	5	0.1467	0.06663	0.5458	32	5	0.07727	0.01316	0.8297

Table 14: Complete paired results for Criteo shard 1.

Rows	Blk	Risk	Method	Reduction	Rows	Blk	Risk	Method	Reduction
12	1	0.1985	0.06101	0.6926	24	1	0.09741	0	1.000
12	2	0.1912	0.05427	0.7162	24	2	0.09967	0.02347	0.7646
12	3	0.2	0.06667	0.6667	24	3	0.09894	0.01982	0.7997
12	4	0.2032	0.05296	0.7394	24	4	0.1034	0.02586	0.75
12	5	0.2105	0.04912	0.7667	24	5	0.09662	0.01659	0.8283
16	1	0.1436	0.05452	0.6203	32	1	0.07378	0.01645	0.7771
16	2	0.1611	0.0487	0.6977	32	2	0.07371	0	1.000
16	3	0.1533	0.06234	0.5932	32	3	0.07424	0.01817	0.7553
16	4	0.1485	0.064	0.5691	32	4	0.07277	0	1.000
16	5	0.1432	0.05438	0.6202	32	5	0.07376	0.02014	0.727

random-only optimised starts show that the primary start still contributes on some blocks. Table 16 and Figure 7 report complementary initialisation controls.

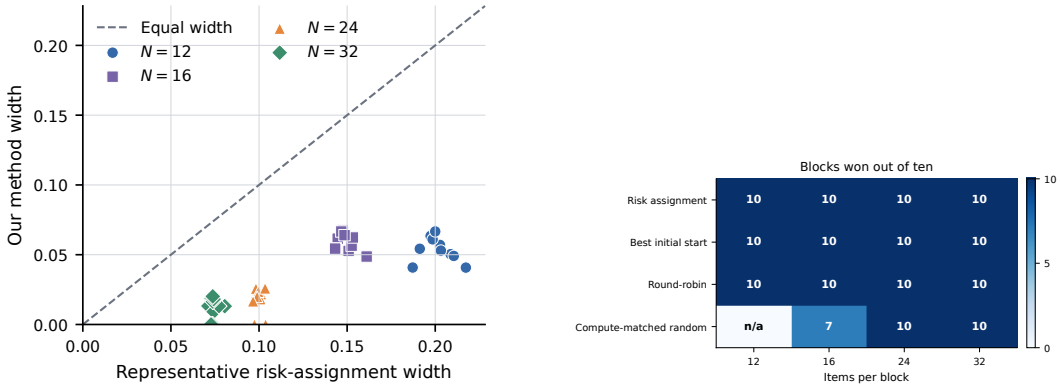

Figure 6: Left: paired widths on the 40-block Criteo grid, with every method width below its scale-specific risk reference. Right: strict method wins out of ten blocks at each scale. The compute-matched entry is absent at twelve rows because that partition space is enumerated.

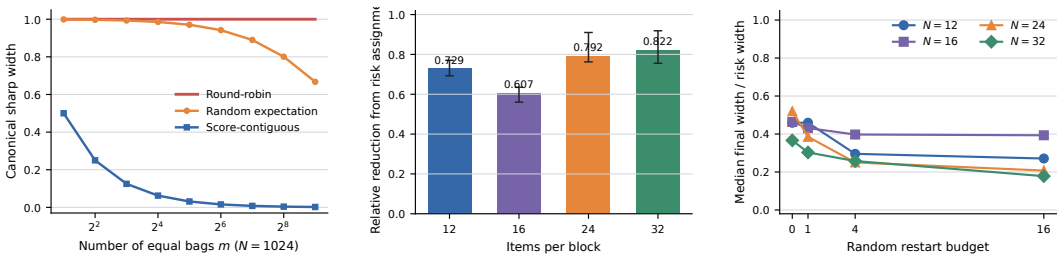

Figure 7: Left: canonical AUC-information spectrum when $n\pi$ is integral, where contiguous width is $1/m$ and round-robin width is one. Middle: block-level relative reduction by scale on the Criteo grid. Right: restart ablation, showing median final width normalised by the risk-reference width.

C.6 Detailed Main-Text Comparisons

The following tables retain the detailed assignment widths and search comparisons discussed in Section 6.

Table 15: **Adjacent-rank grouping sharply reduces AUC ambiguity at equal feedback budgets.** All assignments use $4{,}096$ bags, with Criteo and Adult evaluated on shard 0. The random column gives the range across ten assignments for Criteo and the median across ten assignments for the other workloads. The contiguous and round-robin widths are deterministic. Downward arrows mean smaller widths are better, and bold marks the smallest width in each row.

Workload	Capacity	Contiguous ↓	Random ↓	Round-robin ↓
Criteo	256	**0.0002395**	0.99226–0.99248	0.99614
Adult	4	**0.0001413**	0.6292	0.7964
UNSW-NB15	128	4.72×10^{-5}	0.954	0.9614

Table 16: **Swaps and additional starts both reduce width on Criteo.** Left columns count strict wins out of ten blocks at each scale, so larger counts are better. The joint row counts blocks where the method beats every listed comparator. Right columns independently vary the number of random starts added to the primary start and give the median final width divided by the risk-reference width. The row labels apply to the left columns only, so a left entry and the right entry beside it describe unrelated comparisons. Smaller ratios mean narrower intervals. The dash marks the enumerated scale.

Strict wins out of ten blocks					Effect of added starts	
Comparison	12	16	24	32	Random starts	Median width / risk
Risk assignment	10	10	10	10	0	0.4530
Best initial start	10	10	10	10	1	0.4313
Round-robin	10	10	10	10	4	0.2784
Compute-matched random	–	7	10	10	16	0.2474
All comparisons jointly	–	7	10	10		

C.7 Uncertainty of the Width Comparisons

Table 17 reports percentile bootstrap intervals for paired block-level reductions under the label-derived reference worlds. Table 18 reports uncertainty of the random-assignment medians used in the additional-workload rows of Table 1.

Table 17: Median paired reductions and 95% descriptive percentile bootstrap intervals over ten blocks per row, using 10,000 resamples. Larger reductions are better.

Workload	Rows	Median reduction	Lower	Upper
Criteo	12	0.7291	0.6926	0.7703
Criteo	16	0.6067	0.5603	0.6353
Criteo	24	0.7925	0.7623	0.91
Criteo	32	0.8218	0.7553	0.9186
Adult	12	0.7314	0.6862	0.8029
Adult	16	0.7656	0.7084	0.8518
Adult	24	0.8835	0.8416	0.9487
Adult	32	0.8428	0.8282	0.8531

Table 18: Random-assignment medians with 95% percentile bootstrap intervals over ten random assignments, using 10,000 resamples, at the main-table capacities. Smaller widths are better.

Workload	Capacity	Median	Lower	Upper
Adult	4	0.6292	0.6259	0.6324
UNSW-NB15	128	0.954	0.9538	0.9542

C.8 Size of the Coupling Term

This subsection quantifies the coupling term of Section 5.1 on the same exact-repeat blocks. For each block and each design we evaluate the coupled width of Proposition 5 and the separable sum that treats every occurrence of an atom as its own coordinate, under one reference world and one budget vector. Table 19 pools the census and click-log blocks at each size and reports the median of their ratio with a 95% percentile bootstrap interval over blocks. Its Narrower column counts the blocks whose coupled width is smaller by more than 10^{-10}, and its Shared column gives the range of the number of atoms that the design spreads over more than one bag.

Score-contiguous and risk-based assignment leave every atom inside a single bag at all four sizes, and the two widths then agree to within 9×10^{-16} on each of the 80 blocks. The width-directed assignments place between 3 and 15 atoms across bags and are strictly narrower everywhere. At twelve rows those assignments are global minimisers obtained by complete enumeration, and Appendix C.9 enumerates the sixteen-row minimum as well, while the twenty-four and thirty-two row entries use saved search results that carry no global certificate. Over the 240 endpoint evaluations the largest excess of the coupled width above the separable sum was 9×10^{-17} and the largest primal–dual objective difference was 9×10^{-16}. Each interval resamples this fixed block set rather than an independent source population. The three designs run on the same 80 blocks, so the twelve rows of the table are not twelve separate samples.

Table 19: Coupled width relative to the separable sum on the exact-repeat blocks, twenty blocks per row of the table. Smaller ratios mean a larger coupling term.

Design	Rows	Median ratio	Lower	Upper	Narrower	Shared
Score-contiguous	12	1	1	1	0	0
Score-contiguous	16	1	1	1	0	0
Score-contiguous	24	1	1	1	0	0
Score-contiguous	32	1	1	1	0	0
Risk-based	12	1	1	1	0	0
Risk-based	16	1	1	1	0	0
Risk-based	24	1	1	1	0	0
Risk-based	32	1	1	1	0	0
Width-directed	12	0.104	0.08055	0.1117	20	3
Width-directed	16	0.111	0.08728	0.1357	20	3–6
Width-directed	24	0.02663	0.01742	0.03466	20	6–12
Width-directed	32	0.02565	0.02124	0.02983	20	6–15

C.9 Complete Enumeration at Sixteen Rows

This subsection certifies the attainable width minimum one scale above the twelve-row enumeration. Each sixteen-row block admits $2{,}627{,}625$ balanced capacity-four partitions, which is 455 times the twelve-row count. The coupled width of Proposition 5 depends on a partition only through the bag–atom incidence matrix, so the $45{,}885$ distinct matrices carry every attainable width and the paired endpoint programs run once for each of them. Solving the decoded minimiser again with the unreduced program reproduces every certified value to within 6×10^{-17}, and the score-contiguous width recomputed this way matches the recorded risk reference in all twenty blocks.

Table 20 lists the certified minima beside the score-contiguous reference and the saved search width. Direct width minimisation is narrower than score-contiguous grouping in every block, with median paired reductions 0.636 on the click log and 0.793 on the census workload. The certified coupled-to-separable ratio has median 0.1057 over the twenty blocks, below the saved-search value in Table 19. The local search of Section 5.2 returns the certified minimum in twelve of the twenty blocks, so the sixteen-row search widths reported elsewhere are upper bounds on the attainable minimum rather than the minimum itself.

Table 20: Certified sixteen-row minima from complete enumeration of all 2,627,625 balanced partitions per block. SC is score-contiguous grouping and Search is the saved local-search width. Smaller widths are better, and the last column divides the certified width by the separable sum at the same assignment.

Workload	Shard	Block	SC ↓	Search ↓	Certified ↓	Ratio
Adult	0	0	0.14234	0.04150	**0.03025**	0.0874
Adult	0	1	0.16217	0.02373	**0.02373**	0.0661
Adult	0	2	0.13303	0.03568	**0.03568**	0.0997
Adult	0	3	0.18360	0.05614	**0.05614**	0.1471
Adult	0	4	0.16312	0.03272	**0.03272**	0.0858
Adult	1	0	0.17643	0.02774	**0.02774**	0.0734
Adult	1	1	0.17370	0.02213	**0.02213**	0.0568
Adult	1	2	0.16543	0.05127	**0.04513**	0.1117
Adult	1	3	0.16586	0.02458	**0.02458**	0.0526
Adult	1	4	0.19139	0.05198	**0.05198**	0.1385
Criteo	0	0	0.15085	0.05273	**0.05273**	0.1246
Criteo	0	1	0.15254	0.05647	**0.04107**	0.0974
Criteo	0	2	0.14760	0.06727	**0.05422**	0.1304
Criteo	0	3	0.14466	0.06149	**0.06149**	0.1415
Criteo	0	4	0.14670	0.06663	**0.05579**	0.1353
Criteo	1	0	0.14357	0.05452	**0.05452**	0.1351
Criteo	1	1	0.16107	0.04870	**0.04870**	0.0823
Criteo	1	2	0.15326	0.06234	**0.05541**	0.1283
Criteo	1	3	0.14851	0.06400	**0.05081**	0.1209
Criteo	1	4	0.14317	0.05438	**0.05237**	0.0979

C.10 Feature-Value Bag Construction

This subsection gives the method-level reference of Section 6.2 that builds bags from one observed feature instead of the scorer output. Curated-bag construction groups rows that share a selected feature value and leaves the bag sizes to the data (Chen et al., 2023), so we match the released bag count by sorting the evaluation rows on that feature and cutting the sorted order into equal-capacity bins. Stable original row order breaks feature ties. Each workload uses the first feature listed in its scorer configuration. That is `age` on the census workload, `sbytes` on the network workload and `integer_feature_1` on the click log. The feature bins reuse the row set, the capacity and the bag count of adjacent-rank grouping under the same reference outcome vector. The widths in Table 21 therefore pair with the columns beside them.

Across all 13 capacity settings the feature bins are narrower than the balanced-random median and wider than adjacent-rank grouping. At the capacities of Table 1 the ratio of the feature-bin width to the adjacent-rank width is 3053 on the census workload, 5614 on the network workload and 3731 on the click log. The comparison concerns the width of the AUC interval under a fixed scorer rather than the behaviour of such bags during training.

Table 21: Feature-value bins beside the score-based designs at matched bag counts. SC is adjacent-rank grouping, Feature is the feature-value binning described above, Random is the median over ten balanced random assignments and R-robin is rank-spreading round-robin assignment. Smaller widths are better.

Workload	Capacity	SC ↓	Feature ↓	Random ↓	R-robin ↓
Adult	4	**0.0001413**	0.4313	0.6292	0.7964
Adult	32	**0.001387**	0.6531	0.9436	0.9786
Adult	64	**0.002835**	0.6735	0.9707	0.9896
Adult	128	**0.005719**	0.6838	0.9852	0.9952
Adult	256	**0.01149**	0.6890	0.9928	0.9972
UNSW-NB15	32	1.05×10^{-5}	0.2046	0.9410	0.9363
UNSW-NB15	64	1.78×10^{-5}	0.2324	0.9485	0.9498
UNSW-NB15	128	4.72×10^{-5}	0.2648	0.9540	0.9614
UNSW-NB15	256	8.65×10^{-5}	0.3130	0.9584	0.9619
Criteo	32	2.91×10^{-5}	0.8249	0.9413	0.9688
Criteo	64	5.92×10^{-5}	0.8617	0.9701	0.9843
Criteo	128	**0.0001193**	0.8825	0.9849	0.9921
Criteo	256	**0.0002395**	0.8937	0.9924	0.9961

C.11 Agreement with the Fréchet Bound

This subsection evaluates the bound of Polo et al. (2024) on our own designs, as reported in Section 6.2. We take the bag index as the conditioning variable and the affine AUC integrand of Proposition 1 as the bounded objective, then solve the dual of that bound exactly by threshold search over the integer average-rank numerators at each bag budget. The integer sums are carried without rounding. Over 169 evaluations the largest deviation between the bound width and the row-coordinate width of Theorem 2 is zero. Independent row-variable programs on the full census window and 120 stored entries from the panels above reproduce the same widths with a largest deviation of zero as well. The widths reported for separate rows are therefore this bound evaluated at our partitions, computed in closed form.

Repeated observable states carry information that bag marginals alone do not. Table 22 evaluates both quantities on one partition of each window, holding the partition and the capacities and the reference probabilities fixed so that only the constraint set changes. Adding the equalities that tie the occurrences of one state together narrows the interval at every capacity. The two columns rest on different information about the response probabilities rather than on different estimators under the same information, since the marginal column does not require occurrences of one state in different bags to carry the same probability.

Table 22: Marginal and coupled widths on one shared partition of each window at capacities 32 to 256. The marginal column enforces the bag means alone and the coupled column adds the equalities of Proposition 5. Smaller widths are better.

Window	Capacity	Marginal ↓	Coupled ↓
Census	32	1.111×10^{-3}	6.105×10^{-4}
Census	64	2.213×10^{-3}	1.429×10^{-3}
Census	128	4.240×10^{-3}	3.384×10^{-3}
Census	256	8.385×10^{-3}	7.603×10^{-3}
Network	32	1.299×10^{-4}	1.842×10^{-5}
Network	64	2.200×10^{-4}	3.892×10^{-5}
Network	128	3.665×10^{-4}	8.208×10^{-5}
Network	256	5.887×10^{-4}	1.756×10^{-4}

C.12 Full-Window Repeated-State Curves

This subsection carries the repeated-state comparison of Section 6.4 to the full evaluation windows. The census window keeps the longest capacity-compatible prefix of its evaluation pool, which is

25,856 of 26,049 rows or a share of 0.9926. The network window keeps all 524,288 rows. The row set is the same at every capacity. The bag–state counts of Proposition 5 enter as a sparse matrix, and each candidate solves $\min$ and $\max$ of $c^\top \widetilde{u}$ subject to $A_B \widetilde{u} = A_B u^0$ and $0 \le \widetilde{u} \le 1$ under full-evaluation-pool Jeffreys-smoothed rates. Here states use exact equality of the complete preprocessed scorer feature vectors, including singleton states; the small-block study uses raw numeric keys. The risk reference is score-contiguous grouping, which minimises the equal-capacity within-bag squared-score objective at these window sizes.

Width search at these sizes uses a fixed candidate budget in place of the exhaustive row-exchange neighbourhood of Section 5.2. Each window and capacity evaluates 34 candidate partitions, beginning at the risk reference and proposing interleavings of neighbouring bag groups and batch exchanges between neighbouring bags at boundary and interior positions. The candidate order, the exchange counts and the proposal seeds are set before execution. Each round starts from the narrowest partition of the previous round, and only strict improvements are kept. The search width is therefore the smallest width among the evaluated candidates, and the difference in Table 23 is the improvement reachable within that budget rather than a certified distance to the narrowest balanced partition. The saved final result contains 408 candidate-width evaluations over twelve window–capacity cells, including four bridge-window cells omitted from Table 23. Two initial solve failures were recovered, leaving no unresolved failures; the largest final equality residual is 1.194×10^{-12}.

Table 23 reports a positive difference at every capacity on both windows, so the width objective and the risk objective select different bags at these window sizes as well as on the enumerable blocks. The quantity varied along the table is bag capacity at a fixed number of rows, so the table describes the capacity range covered and not a limit in the number of rows.

Table 23: Shared-state full windows: Adult has 25,856 rows and UNSW-NB15 has 524,288. Each capacity evaluates 34 predefined proposals under label-derived smoothed reference probabilities. Reduction is the paired gap divided by the risk width, computed before rounding.

Window	Capacity	Risk	Search	Gap	Reduction (%)
Adult	32	6.42×10^{-4}	6.10×10^{-4}	3.11×10^{-5}	4.84
Adult	64	1.50×10^{-3}	1.43×10^{-3}	6.98×10^{-5}	4.66
Adult	128	3.49×10^{-3}	3.38×10^{-3}	1.06×10^{-4}	3.03
Adult	256	7.71×10^{-3}	7.60×10^{-3}	1.11×10^{-4}	1.44
UNSW-NB15	32	1.87×10^{-5}	1.84×10^{-5}	2.80×10^{-7}	1.50
UNSW-NB15	64	3.96×10^{-5}	3.89×10^{-5}	6.58×10^{-7}	1.66
UNSW-NB15	128	8.35×10^{-5}	8.21×10^{-5}	1.44×10^{-6}	1.73
UNSW-NB15	256	1.77×10^{-4}	1.76×10^{-4}	1.79×10^{-6}	1.01

D Complete Results on the Census and Network Workloads

The tables in this appendix carry every cell summarised in Section 6. The scorer and preprocessing follow the specified Criteo configuration. The assignment methods and random seeds also follow that configuration, as do the block grid and search settings. Only the rows change.

D.1 Assignment Spectrum

This subsection gives the census and network assignment comparisons in Section 6.2.

Table 24 gives the complete capacity comparison on both census shards. Table 25 gives the corresponding capacity comparison on the network workload. Each random column is the median of ten balanced random assignments, and each ordering column counts seeds with contiguous $<$ random $<$ round-robin. Every observed contiguous width in these comparisons stays below $1/m$.

D.2 Independent-Random-Score Comparison

This subsection supports the rank-alignment comparison in Section 6.2.

Table 26 lists the mean paired difference between the independent-random-score comparison and a balanced random assignment in each of the 21 cells. The largest absolute value is 5.41×10^{-3},

Table 24: Assignment spectrum on the census workload. Rows are the evaluation windows of the two Adult shards under the row-coordinate closed form.

Shard	Capacity	Bags	Contiguous	Random median	Round-robin	Ordering
0	4	4096	0.0001413	0.6292	0.7964	10/10
0	8	2048	0.0003199	0.7946	0.9012	10/10
0	16	1024	0.0006782	0.8916	0.951	10/10
0	32	512	0.001387	0.9436	0.9786	10/10
0	64	256	0.002835	0.9707	0.9896	10/10
0	128	128	0.005719	0.9852	0.99519	10/10
0	256	64	0.01149	0.99279	0.99719	10/10
1	4	2048	0.0002804	0.6281	0.7997	10/10
1	8	1024	0.0006187	0.7884	0.9038	10/10
1	16	512	0.001341	0.8867	0.9512	10/10
1	32	256	0.002761	0.9407	0.9753	10/10
1	64	128	0.005703	0.9703	0.99067	10/10
1	128	64	0.01145	0.9846	0.99568	10/10
1	256	32	0.02293	0.99255	0.99720	10/10

Table 25: Assignment spectrum on a network workload matched to the scale of the advertising shards. The evaluation window holds $524{,}288$ rows, so capacity 128 reproduces the $4{,}096$ bags of the main setting. Scores are heavily tied here and the round-robin column is not the largest entry at capacity 32.

Capacity	Bags	Contiguous	Random median	Round-robin	Ordering
4	131,072	6.60×10^{-7}	0.7446	0.7593	10/10
8	65,536	1.34×10^{-6}	0.8672	0.9363	10/10
16	32,768	3.47×10^{-6}	0.9214	0.9363	10/10
32	16,384	1.05×10^{-5}	0.941	0.9363	0/10
64	8,192	1.78×10^{-5}	0.9485	0.9498	10/10
128	4,096	4.72×10^{-5}	0.954	0.9614	10/10
256	2,048	8.65×10^{-5}	0.9584	0.9619	10/10

which is 6.86×10^{-3} of the contiguous-to-random gap that the comparison addresses. Sixteen of the 21 descriptive intervals include zero, and the five that exclude it have both signs.

Table 26: Independent-random-score minus balanced-random width. Each cell pairs ten assignments; brackets are saved 95% percentile bootstrap intervals for the mean paired difference. Difference and interval columns are in 10^{-3} AUC-width units. The last column divides the mean difference by the random-median-minus-contiguous gap.

Workload	Shard	Capacity	Difference $\times 10^3$	95% interval $\times 10^3$	Gap (%)
Adult	0	4	2.55	[-2.20, 7.14]	0.40
Adult	0	8	-1.21	[-3.23, 1.00]	-0.15
Adult	0	16	-2.15	[-3.91, -0.47]	-0.24
Adult	0	32	-1.18	[-3.47, 0.89]	-0.13
Adult	0	64	0.25	[-1.52, 2.17]	0.03
Adult	0	128	0.79	[-0.34, 2.07]	0.08
Adult	0	256	0.47	[-0.51, 1.41]	0.05
Adult	1	4	1.88	[-4.53, 8.14]	0.30
Adult	1	8	5.41	[-1.79, 11.68]	0.69
Adult	1	16	5.07	[0.72, 8.31]	0.57
Adult	1	32	2.62	[-0.38, 5.19]	0.28
Adult	1	64	0.57	[-1.54, 2.77]	0.06
Adult	1	128	0.64	[-0.50, 1.72]	0.07
Adult	1	256	0.14	[-0.88, 1.17]	0.01
UNSW-NB15	0	4	-0.81	[-3.27, 1.29]	-0.11
UNSW-NB15	0	8	-0.97	[-2.03, 0.11]	-0.11
UNSW-NB15	0	16	-0.77	[-1.26, -0.27]	-0.08
UNSW-NB15	0	32	-0.35	[-0.70, -0.03]	-0.04
UNSW-NB15	0	64	-0.07	[-0.28, 0.14]	-0.01
UNSW-NB15	0	128	-0.10	[-0.34, 0.15]	-0.01
UNSW-NB15	0	256	-0.20	[-0.32, -0.07]	-0.02

D.3 SHARED-ATOM GRID

This subsection supplies the census search results summarised in Section 6.4.

Table 27 reports the census grid in the same format as the advertising grid. Absolute columns are medians across ten capacity-four blocks per scale, five from each shard, and intervals are 95% descriptive bootstrap summaries of relative reduction. Table 28 places the compute-matched counts of both domains side by side.

Table 27: Direct sharp-width minimisation on the census shared-atom grid. The last two columns count blocks in which the method is strictly narrower than the risk reference and than the best unoptimised partition among the identical starts.

Rows	Bags	Risk width	Method width	Reduction	95% interval	Vs. risk	Vs. starts
12	3	0.1778	0.04475	0.7314	[0.6862, 0.8029]	10/10	10/10
16	4	0.1656	0.0342	0.7656	[0.7084, 0.8518]	10/10	10/10
24	6	0.1039	0.01201	0.8835	[0.8416, 0.9487]	10/10	10/10
32	8	0.07804	0.01284	0.8428	[0.8282, 0.8531]	10/10	10/10

Table 28: Compute-matched random search on both shared-atom domains. Entries are win–tie–loss counts over the ten blocks of each scale, with ties in their own position. The twelve-row scale enumerates its complete partition space instead of sampling it, so no matched search is defined there.

Rows	Bags	Criteo	Adult	Shard	Criteo median	Adult median
12	3	–	–	0	0.7666	0.8280
16	4	7–3–0	8–2–0	1	0.7447	0.8334
24	6	10–0–0	10–0–0			
32	8	10–0–0	10–0–0			

The paired reductions and absolute widths in Table 27 show the effect at each scale without pooling blocks of different sizes.

D.4 Observable States in Covertype

This subsection illustrates the repeated-state condition in Section 5.1.

Covertype illustrates the condition required by the shared-atom construction through the multiplicity of its observed feature states. Its 581,012 rows produce 581,012 distinct values of the ten quantitative features. No observable state therefore repeats, and the maximum multiplicity is one. The coupled programs consequently have no shared state to couple across bags. The row-coordinate theory of Section 4 still applies to that workload. Only the shared-atom study is empty there, which places Covertype and the census table at the two ends of the coverage range in Table 8.

D.5 Agreement between the Closed Form and the Programs

This subsection supports the scorer and endpoint comparisons in Section 6.4.

Table 34 collects numerical consistency checks separately from design performance. The nonlinear scorer comparison in Table 29 retains the separate-row assignment ordering across both Adult shards. The theory operates on fixed scores, so its role is to check a second empirical rank source.

Table 29: Adult with the fixed two-layer scorer, separate row probabilities and the realised outcome reference. Random entries are medians over ten assignments. Ordering counts contiguous $<$ random $<$ round-robin in paired assignments.

Shard	Capacity	Contiguous	Random	R-robin	Ordering
0	4	1.24×10^{-4}	0.6347	0.8045	10/10
0	8	3.00×10^{-4}	0.7973	0.9084	10/10
0	16	6.38×10^{-4}	0.8926	0.9573	10/10
0	32	1.29×10^{-3}	0.9438	0.9797	10/10
0	64	2.65×10^{-3}	0.9706	0.9895	10/10
0	128	5.31×10^{-3}	0.9852	0.9942	10/10
0	256	0.0107	0.9928	0.9972	10/10
1	4	2.48×10^{-4}	0.6333	0.7976	10/10
1	8	5.91×10^{-4}	0.7908	0.9017	10/10
1	16	1.28×10^{-3}	0.8868	0.9517	10/10
1	32	2.65×10^{-3}	0.9419	0.9768	10/10
1	64	5.36×10^{-3}	0.9705	0.9873	10/10
1	128	0.01086	0.9846	0.9934	10/10
1	256	0.02182	0.9921	0.9974	10/10

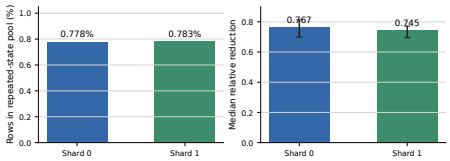

Figure 8: Block-level reductions by advertising shard, with corresponding census medians 0.8280 and 0.8334.

D.6 Flat-World Predictions and Observed Widths

Table 30 compares the canonical prediction with the observed width for every capacity setting. Systematic deviations on Adult and UNSW-NB15 arise from non-flat reference worlds and tied scores, which differ from the canonical assumptions.

Table 30: Canonical flat-world diagnostic at each observed prevalence. Predictions assume distinct scores and use the fractional correction in Proposition 3; observations use non-flat outcome references and may have ties. Relative deviations are $(\text{observed} - \text{predicted})/\text{predicted}$ in percent. Random compares its observed mean over ten assignments with the theoretical expectation.

Workload	Shard	Capacity	SC pred.	SC obs.	SC (%)	Random (%)	RR (%)
Criteo	0	256	2.44×10^{-4}	2.40×10^{-4}	-1.9	-0.4	-0.4
Adult	0	4	2.40×10^{-4}	1.41×10^{-4}	-41.1	-20.0	-19.0
Adult	0	8	4.84×10^{-4}	3.20×10^{-4}	-34.0	-10.0	-9.2
Adult	0	16	9.73×10^{-4}	6.78×10^{-4}	-30.3	-4.9	-4.6
Adult	0	32	1.95×10^{-3}	1.39×10^{-3}	-28.9	-2.6	-2.0
Adult	0	64	3.91×10^{-3}	2.83×10^{-3}	-27.4	-1.4	-1.0
Adult	0	128	7.81×10^{-3}	5.72×10^{-3}	-26.8	-0.7	-0.5
Adult	0	256	0.01562	0.01149	-26.5	-0.4	-0.3
Adult	1	4	4.78×10^{-4}	2.80×10^{-4}	-41.3	-19.7	-18.3
Adult	1	8	9.67×10^{-4}	6.19×10^{-4}	-36.0	-10.3	-8.7
Adult	1	16	1.95×10^{-3}	1.34×10^{-3}	-31.1	-5.4	-4.5
Adult	1	32	3.90×10^{-3}	2.76×10^{-3}	-29.2	-2.7	-2.3
Adult	1	64	7.81×10^{-3}	5.70×10^{-3}	-27.0	-1.4	-0.9
Adult	1	128	0.01562	0.01145	-26.7	-0.7	-0.4
Adult	1	256	0.03125	0.02293	-26.6	-0.4	-0.3
UNSW	0	4	5.79×10^{-6}	6.58×10^{-7}	-88.6	22.6	0.0
UNSW	0	8	1.35×10^{-5}	1.34×10^{-6}	-90.1	10.2	5.7
UNSW	0	16	2.90×10^{-5}	3.47×10^{-6}	-88.0	3.2	-1.3
UNSW	0	32	5.98×10^{-5}	1.05×10^{-5}	-82.5	-1.0	-4.5
UNSW	0	64	1.22×10^{-4}	1.78×10^{-5}	-85.4	-3.3	-4.7
UNSW	0	128	2.44×10^{-4}	4.72×10^{-5}	-80.7	-3.7	-3.7
UNSW	0	256	4.88×10^{-4}	8.65×10^{-5}	-82.3	-3.8	-3.8

D.7 Reference-World Sensitivity

This comparison changes the probabilities used to select bags while evaluating all selected assignments under the same label-derived reference world. Label-derived probabilities use evaluation-shard responses. Predicted probabilities use the fixed scorer, constant probabilities equal 0.5, and the Adult training-rate reference uses fit-split Jeffreys rates for each observed numeric state, with probability 0.5 for unseen states. Table 31 reports paired reductions and their descriptive block-bootstrap intervals. Alternative references preserve positive median improvements while changing many selected partitions (Table 32). This establishes sensitivity to design information under a common evaluation world; it does not replace the population mean assumption with noisy empirical feedback.

Table 31: Design-reference sensitivity. All returned assignments are evaluated in the same label-derived world. Each row uses ten blocks; reduction is the median paired percentage and brackets are saved 95% block-bootstrap intervals. W/T/L compares search with the risk reference in that evaluation world.

Workload	Rows	Design reference	Reduction (%)	95% interval	W/T/L
Criteo	12	Label-derived	72.9	[69.3, 77.0]	10/0/0
Criteo	12	Predicted	65.5	[58.0, 72.7]	10/0/0
Criteo	12	Constant 0.5	67.3	[55.7, 71.6]	10/0/0
Criteo	16	Label-derived	60.7	[56.9, 63.5]	10/0/0
Criteo	16	Predicted	56.3	[50.5, 61.4]	10/0/0
Criteo	16	Constant 0.5	58.7	[54.6, 61.8]	10/0/0
Criteo	24	Label-derived	79.2	[76.2, 91.0]	10/0/0
Criteo	24	Predicted	71.7	[66.3, 75.5]	10/0/0
Criteo	24	Constant 0.5	71.9	[68.8, 86.2]	10/0/0
Criteo	32	Label-derived	82.2	[75.5, 91.9]	10/0/0
Criteo	32	Predicted	100.0	[76.8, 100.0]	10/0/0
Criteo	32	Constant 0.5	79.9	[61.6, 100.0]	10/0/0
Adult	12	Label-derived	73.1	[68.6, 80.3]	10/0/0
Adult	12	Predicted	73.1	[68.2, 80.3]	10/0/0
Adult	12	Constant 0.5	68.4	[60.9, 78.3]	10/0/0
Adult	12	Training-rate	68.5	[62.6, 80.3]	10/0/0
Adult	16	Label-derived	76.6	[70.8, 85.2]	10/0/0
Adult	16	Predicted	70.4	[66.6, 72.1]	10/0/0
Adult	16	Constant 0.5	70.8	[69.7, 73.9]	10/0/0
Adult	16	Training-rate	72.0	[68.4, 84.3]	10/0/0
Adult	24	Label-derived	88.4	[84.2, 94.9]	10/0/0
Adult	24	Predicted	81.6	[78.0, 85.2]	10/0/0
Adult	24	Constant 0.5	77.4	[71.5, 79.7]	10/0/0
Adult	24	Training-rate	79.3	[67.3, 92.9]	10/0/0
Adult	32	Label-derived	84.3	[82.8, 85.3]	10/0/0
Adult	32	Predicted	76.7	[70.6, 100.0]	10/0/0
Adult	32	Constant 0.5	75.0	[59.0, 90.4]	10/0/0
Adult	32	Training-rate	93.6	[76.1, 100.0]	10/0/0

Table 32: Changes in atom-to-bag membership relative to label-derived design. Each comparison reuses the same forty blocks.

Workload	Reference	Changed / blocks
Criteo	predicted	35/40
Criteo	constant	30/40
Adult	predicted	30/40
Adult	constant	34/40
Adult	training rate	29/40

D.8 Larger Repeated-State Blocks

The extension in Section 6.4 uses disjoint Adult shard-0 blocks at capacity four with two occurrences per atom. It starts from contiguous grouping and samples cross-bag swaps, accepting the first strict width improvement until the 10,000-proposal or 100-accepted-swap limit is reached. Every reported run stops at the proposal budget. The budget is 10,000 candidate evaluations per block and comparisons use the same label-derived reference probabilities as the smaller-block study. Table 33 gives the scale-wise medians.

Table 33: Sampled width descent on larger Adult blocks. Widths are smaller under direct descent in every block.

Rows	Blocks	Contiguous median	Search median	Median reduction
64	5	0.03687	0.004606	0.8622
128	5	0.01875	0.004176	0.7704
256	5	0.008761	0.004471	0.4975

E Reproduction Details

This appendix specifies the implementation of the experiments in Section 6.

E.1 End-to-End Sequence

This subsection supplies the implementation steps for the comparisons in Section 6.

The study has four stages. The first stage loads the declared chronological data files, fits the numeric logistic scorer on the leading block and scores the following rows with it. The row-coordinate stage forms the score-contiguous and round-robin assignments together with ten balanced random and ten independent-random-score assignments, treats rows as separate latent coordinates and computes the closed-form endpoints independently for every bag. The shared-atom stage groups evaluation rows by exact equality of the selected raw numeric feature keys, filters the resulting atoms by multiplicity and applies the fixed label-independent order to their feature keys. It retains the leading two rows in stable order and computes the smoothed reference probabilities from the full atom multiplicities. The comparison stage allocates the objective-gap set followed by the disjoint method-calibration set and then the final shard-by-scale grid, applies complete enumeration at twelve rows and direct width search on the final grid, and aggregates the block-level outputs.

E.2 Feasible Assignments and Endpoint Computation

This subsection specifies the feasible assignments and endpoint calculations used in Section 6.4.

Table 34 separates numerical agreement from assignment performance. The closed-form and LP comparisons use the same constraints.

Table 34: Numerical and channel checks under the stated probability model. Differences are absolute AUC-width errors. Witness-blind entries are remaining identification widths, not errors.

Check	Evaluation set	Maximum error / width
Separable versus LP	Criteo	3.33×10^{-16}
Separable versus LP	Adult linear, 120 configurations	1.28×10^{-16}
Separable versus LP	Adult nonlinear, 120 configurations	1.18×10^{-16}
Separable versus LP	UNSW, 60 configurations	1.11×10^{-16}
Fréchet versus separable	169 evaluations	0
Canonical random expectation	Pools up to 1,000; 400 partitions	< 0.003
Witness-blind remaining width	Criteo, same 22 assignments	1
Witness-blind remaining width	Adult shards 0 / 1	0.999991 / 1
Witness-blind remaining width	UNSW	0.962223

E.3 Numerical Conventions

This subsection specifies the numerical settings for Section 5.

Endpoints and search decisions. Ranks use half credit for score ties through (3), and widths use double precision. Both endpoint solves must report success through SciPy's HiGHS route, using its default solver tolerances. A solve failure terminates the reference-world calculation. Strict width comparisons and numerical ties use 10^{-12} before display rounding. Residuals recorded by the supplementary evaluator are diagnostics, not an additional 10^{-12} acceptance rule.

Result summaries. Bootstrap intervals use 10,000 percentile resamples of the stated experimental units. Medians of reductions are computed from paired block reductions, not ratios of pooled widths. These intervals describe variation across the observed blocks and do not confer confidence coverage on the sharp identification endpoints.

Software. The implementation uses Python 3.10.16, NumPy 2.2.6, pandas 2.3.1, PyArrow 19.0.1, scikit-learn 1.7.1 and SciPy 1.15.3.

E.4 Randomness and Execution

This subsection supplies the randomisation and computing settings for Section 6.1.

We specify the random seeds for each comparison below. The ten full-shard balanced assignments use seeds 830311 to 830320 and their independent-random-score assignments use 830321 to 830330, with the same seeds reused on the census and network workloads. The paired independent-random-score bootstrap uses seed 830399 and the objective-gap bootstrap uses 830421. The ten disjoint method-calibration blocks use restart seeds 830532 to 830541. For the final grid, let $s \in \{0, 1\}$ denote the shard index. Let $r \in \{0, 1, 2, 3\}$ index the ordered row counts $(12, 16, 24, 32)$. The within-scale block index is $k \in \{0, \dots, 4\}$. The restart seed for that block is $830800 + 10000s + 100r + k$. Compute-matched random search uses the corresponding seed plus 1000 at 16 to 32 rows. Scale-level bootstrap seeds are $830640 + r$ and shard-level bootstrap seeds are $830740 + s$. The closed-form comparison against the endpoint programs uses seeds 830341 to 830343, and both scorers use seed 830300.

The advertising grid was evaluated with 32 independent block workers on a host with 128 logical CPUs, and the census grid with 20 workers on a 24-core desktop. Each numerical library used one thread per worker. Parallel scheduling does not enter the scientific comparison, because candidate evaluations rather than wall-clock time define the compute-matched budget. The census computation runs in about twelve seconds for the row-coordinate stage and about one hour for the 40-block shared-atom stage, and the network stage is a row-coordinate study of comparable cost to the census one.

E.5 Design Interpretation and Scope

This subsection expands the interpretation and scope of the design results developed in Sections 4 and 5.

Bag homogeneity is not a complete design principle. What matters is the gradient of the downstream functional over latent worlds. For AUC this gradient follows the rank contrast G. The practical order is therefore to specify the target before its constraints and then choose an assignment. The width we optimise is also not estimation error: it persists under exact population bag means, so repeated noisy observations shrink the error around each bag mean without separating worlds that imply the same exact means. A narrow target interval can therefore reveal more about a declared functional without identifying individual labels, which is what Proposition 8 says.

Several assumptions bound the results. The theory assumes an exact-mean channel. Under empirical binary releases the shared-state exact-mean constraints are often infeasible because realised counts differ from population means. Presence-only and randomised-response channels change the compatible set, as do count-noise and differentially private channels. Equation (4) describes where assignment enters these channels. Their sharp bounds additionally depend on the resulting compatible sets. We design for one fixed rank direction, so a release serving several scorers calls for minimising the worst normalised width over their contrast vectors. Heterogeneous capacities and group constraints would be accepted by the endpoint programs but must also be respected by the swap neighbourhood. Two programs per candidate swap confine direct search to small and medium blocks, and global optimality is verified at twelve and sixteen rows. The three-point spectrum assumes distinct scores, and the UNSW-NB15 capacity-32 cell shows what a dominant tied block does to it. Repeated observable states are frequent on the census table and rare on the click log. Covertype marks the far end of that range. Its $581{,}012$ rows produce as many distinct feature values. No observable state therefore repeats, and the coupled programs have no shared state to couple. The row-coordinate theory still applies to this workload. Finally, the reference probabilities that induce candidate budgets are empirical smoothed rates shared by every candidate, and are neither design-time feedback nor ground truth.

F Extended Related Work

F.1 Learning from Given Aggregate Observations

Learning from label proportions (LLP) methods recover training information through mean maps (Quadrianto et al., 2009) or aggregate prediction losses (Ardehaly & Culotta, 2017), while EasyLLP debiases aggregate supervision for empirical risk minimisation (Busa-Fekete et al., 2023). Bag membership can also improve training. Curation groups common-feature instances to recover informative gradients (Chen et al., 2023), and PriorBoost groups predicted responses across learning iterations (Javanmard et al., 2024b).

Learning from label proportions keeps instance features and replaces individual labels by group sums or proportions. Early formulations train predictors directly from aggregate outputs (Musicant et al., 2007), mean-map methods recover class-level feature moments under shared-distribution conditions (Quadrianto et al., 2009), and latent-label large-margin methods enforce the observed proportions inside each bag (Yu et al., 2013). These formulations establish that aggregate feedback can support instance-level prediction, and in all of them bag membership is part of the observation model.

Subsequent approaches use deep learning or belief propagation to obtain individual predictions (Dulac-Arnold et al., 2019; Havaldar et al., 2024). Likelihood-based estimation (Zhang et al., 2020) and unbiased risk estimation (Wei et al., 2023) extend learning across aggregate observation models.

Modern approaches combine proportion constraints with representation learning or probabilistic inference (Dulac-Arnold et al., 2019; Havaldar et al., 2024). Broader formulations model several aggregate observation channels through likelihood-based estimation (Zhang et al., 2020), and other methods construct unbiased classification risks from aggregate supervision (Wei et al., 2023). Theoretical analyses characterise learnability under mutual-contamination assumptions and favourable proportion-sampling regimes (Scott & Zhang, 2020; Busa-Fekete et al., 2023), establish algorithmic limits and optimistic rates (Saket, 2022; Li et al., 2024), and give nearly optimal square-loss guarantees that extend to general losses (Busa-Fekete et al., 2025; Applebaum et al., 2026). Benchmarks and methodological studies show that bag-generation and evaluation choices materially affect reported predictor quality (Brahmbhatt et al., 2024; Franco et al., 2023), and further analyses compare bag-level with instance-level learning objectives (Javanmard et al., 2024a). LLP theory characterises learnability under contamination assumptions and computational limits for linear thresholds (Scott & Zhang, 2020; Saket, 2022). EasyLLP supports unbiased loss estimation (Busa-Fekete et al., 2023), while subsequent analyses establish optimistic rates and sample guarantees for square and general losses (Li et al., 2024; Busa-Fekete et al., 2025; Applebaum et al., 2026).

Dual-proportion methods select one compatible labelling for classifier training (Ma et al., 2026), whereas we retain every instance-level response world compatible with the exact bag means and optimise the sharp AUC interval of a fixed scorer.

F.2 Designing Bags for Aggregate Supervision

Because predictor quality depends on the supervision, several methods modify the bags themselves. GenBags forms weighted linear combinations of observed bag indicators and MixBag creates overlapping augmented bags (Saket et al., 2022; Asanomi et al., 2023). Other methods use smaller bags or perturb supervision in large bags (Baručić & Kybic, 2022; Kubo et al., 2024), and composite constructions change whether weak predictions can be converted into strong instance predictions (Makhija & Saket, 2025).

Direct curation builds bags from instance structure rather than only transforming existing aggregate observations. Structure-matched bags recover informative gradients for semilinear predictors (Chen et al., 2023), and PriorBoost forms capacity-constrained bags from the current prediction order and updates that design across learning iterations (Javanmard et al., 2024b). We also use scorer-derived order as a design signal for choosing a non-overlapping partition before a single release of exact bag means, while retaining a fixed scorer and optimising the resulting sharp AUC width rather than prediction risk. Training-oriented bag design therefore leaves a different question open, namely how much uncertainty a chosen partition leaves about the fixed scorer's AUC.

F.3 Evaluating Fixed Models without Instance Labels

That question is naturally expressed through the set of instance-level response worlds compatible with the released aggregates. Classical and modern ecological analyses formalise how aggregate observations leave multiple individual-level states feasible (Duncan & Davis, 1953; Jiang et al., 2020), and partial-identification theory distinguishes the sharp identified set implied by the observations and assumptions from computational outer sets and sampling confidence statements (Molinari, 2020). Recent aggregate-data analysis similarly obtains target-specific extrema over a common compatible set (Moon, 2026).

The same set-based view appears in machine-learning evaluation. Recent work computes target extrema under discrete measurement error (Finkelstein et al., 2021) and bounds maximum mean discrepancy over compatible contamination assignments (Nafshi & Makar, 2024). Weak-supervision evaluation derives Fréchet bounds from observable marginals for single-example metrics including accuracy and precision as well as recall and F_1 (Polo et al., 2024). Closest in design, learned instrument partitions reduce conditional treatment-effect bound width while accounting for estimation error (Schweisthal et al., 2025). Our uncertainty set instead comes from exact bag means over a fixed finite evaluation pool, and our non-overlapping partition generates those constraints directly rather than discretising an instrument representation.

F.4 AUC Identification and Rank-Based Partition Design

AUC is a pairwise rank functional closely connected to the Mann–Whitney statistic (Mann & Whitney, 1947; Hanley & McNeil, 1982). At fixed prevalence the affine reduction of Proposition 1 nevertheless turns it into the expectation of a single-example function of the response and the rank contrast, so for separate rows the width of Theorem 2 is a Fréchet bound in the sense of Polo et al. (2024) with the bag index as the conditioning variable. What we add in that regime is the closed form and the choice of the partition that generates the marginals. Repeated observable states go further, because they impose equalities that tie one probability to several bags at once and therefore lie outside any bound formed from bag marginals alone. Related work analyses labelled-data AUC generalisation (Agarwal et al., 2005) or trains rankers from contaminated or partially labelled samples (Xie et al., 2024). For a fixed scorer and exact bag means, we characterise the sharp AUC interval and design the partition to reduce its width.

Because AUC depends on score order, partitions that group nearby ranks form a natural design family. Sorted partitions have a long history in optimal grouping and one-dimensional clustering as well as interval dynamic programming (Fisher, 1958; Wang & Song, 2011; Nielsen & Nock, 2014), with exact accelerations that exploit objective-specific Monge or total-monotonicity structure (Grønlund et al., 2017). Related design problems use quantisation to improve downstream estimation (Farias & Brossier, 2013). These methods provide generic optimisation primitives for ordered partitions. We derive the AUC-width structure induced by exact bag means and characterise when assignment changes information while accounting for repeated-state coupling. This connects aggregate-label curation with metric partial identification and target-specific partition design.

Bag membership therefore becomes a design variable for the identified performance range rather than an input held constant throughout evaluation.