WHAT A REPORTED POOLED AUC IDENTIFIES WHEN SCORES ARE UNAVAILABLE

Anonymous authors
Paper under double-blind review

ABSTRACT

A pooled AUC over a stratified evaluation is often interpreted as evidence that a score discriminates well within strata. In general, however, the pooled AUC does not determine within-stratum discrimination. When per-example scores are unavailable, a reader may observe only the reported AUC, the outcome labels, and the stratum memberships. We study exactly what this information identifies, separating two distinct questions. First, what reference value should the reported AUC be compared against? We define its conditional permutation mean: the expected pooled AUC when labels are permuted within strata while scores remain fixed. This is not a new metric, but a reference value induced by the unreleased scores themselves, and it need not equal 0.5. From annotations alone, we derive sharp extrema for this quantity, attained by rankings that place strata in contiguous blocks. Conditioning additionally on an exact reported AUC yields tighter sharp bounds together with an explicit attaining construction. Second, what does the reported pooled AUC imply about within-stratum discrimination? We characterise exactly when it is compatible with within-stratum AUCs of 0 or 1. On PRM800K, the reported AUC substantially narrows the annotation-only range and can exclude 0.5 as the conditional permutation mean. Yet none of the seven AUCs we recompute from the released data rules out a within-problem AUC below 0.5. On ProcessBench, interpreting the published figure as a standard AUC, scores matching 0.765 at the reported precision attain within-solution AUCs of both 0 and 1. These results separate what a pooled AUC identifies from what it leaves undetermined and specify the per-stratum statistics sufficient to identify each quantity.

1 INTRODUCTION

A benchmark result reaches most readers as a headline metric, and that metric is read as a claim about what a model can do. When the per-example scores behind it are not released, that interpretation cannot be audited directly: the reader is left with the figure, the outcome labels and the design. Take a paper reporting one pooled AUC over an evaluation whose examples fall into strata — sampled answers grouped by problem, questions by subject, sentences by passage. A reader who wants to know what that number says about discrimination *inside* a stratum cannot compute a within-stratum statistic. This paper asks what such a reader can verify from what was released.

The number alone does not settle it. Consider two problems with ten sampled answers each: nine of the answers to the easy problem are correct, and one of the answers to the hard problem is. Give every answer to the easy problem the same high score and every answer to the hard problem the same low score. Inside a problem every answer shares one score, so each within-problem pair receives the tie credit of 0.5 and the within-problem AUC is exactly 0.5; the pooled AUC is 0.90. Many score orderings are consistent with a printed 0.90, and they do not agree about what happens inside a problem.

The example exposes two separate failures of the naive reading. First, even with the within-stratum association removed, the pooled figure need not sit at 0.5, because the cross-stratum comparisons remain; the reference it should be read against is a property of the unreleased scores. Second, the pooled figure need not determine discrimination inside a stratum at all. We therefore ask two identification questions. **Q1.** What reference value should the pooled number be compared against, once within-stratum label association is removed and the unreleased scores are held fixed? That

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.

reference is the figure's *conditional permutation mean* ν: the expected pooled AUC when the scores stay fixed and the outcome labels are permuted inside each stratum. We do not propose it as a new performance metric: it is the value the unreleased scores themselves induce, so it is score-specific and can lie far from 0.5 — it is 0.90 in the example above. **Q2.** Which levels of within-stratum discrimination remain compatible with the same reported number?

The printed figure and the stratum outcome counts constrain the answers to the two questions to very different degrees, and that gap is this paper's subject. On PRM800K (Lightman et al., 2024) a reported 0.941 confines ν to $[0.562, 0.908]$, yet the same figure is matched by a score whose within-problem AUC is 0.175. Neither answer is a proxy for the other.

We answer Q1 by partially identifying ν, and the bounds are sharp: an explicit ranking attains them, so they are not a relaxation. On PRM800K's design they exclude 0.5 for printed values above about 0.87, and they certify exclusion rather than membership: a value outside the interval is impossible, whereas a value inside it need not be attainable. Q2 is not settled by those bounds but by a separate analysis of the same release. For the AUC of 0.765 that Lee et al. (2026) report on ProcessBench MATH (Zheng et al., 2025), read as a standard AUC, the bounds on ν are $[0.434, 0.901]$, while scores matching 0.765 at its printed precision attain within-solution AUCs of 0 and of 1.

Contributions. (i) We formulate what a reported pooled AUC identifies when the scores are unavailable, and solve it: annotation-only extrema (Proposition 1) and, conditioning on the reported figure, sharp bounds proved attainable by an explicit ranking (Theorem 1), extended to tied scores (Corollary 1) and to the rank sums a rounded figure permits. (ii) For within-stratum discrimination we characterise exactly which reported values admit the two extremes (Proposition 2), and construct arrangements realising values between them (Proposition 3). (iii) We audit two public releases: PRM800K, whose released scores let us simulate the no-score setting and check the bounds against the quantities they would hide, and a published ProcessBench figure, where the scores really are unavailable. (iv) We state the per-stratum statistics that would identify each quantity, so a future release can be audited without the scores (Section 6).

2 Related work

AUC decomposition and conditional evaluation. Identity (1) is a regrouping of Proposition 1 of Kallus & Zhou (2019), which decomposes a population AUC into cross-group and within-class terms, and Borkan et al. (2019) score subgroups against pooled negatives for a related purpose; collecting the diagonal terms gives λ and A_{within}. The covariate-adjusted ROC curve (Janes & Pepe, 2009; Inácio & Rodríguez-Álvarez, 2022) targets the within-covariate quantity directly; grouped AUC in recommendation (Zhou et al., 2018) computes one AUC per user and averages, weighting users by impressions rather than by pairs (Appendix G); van Elteren (1960) combines within-stratum rank statistics with fixed weights, and survival analysis restricts the concordance index to within-stratum pairs for the reason (1) gives (Harrell et al., 1996; van Houwelingen & Putter, 2011). Sudjianto & Liu (2025) use the decomposition diagnostically. Both the distinction between the two quantities and the estimation of the within-stratum one when scores are available are established in prior work.

Conditional permutation and evaluation audits. Chaibub Neto et al. (2019) permute labels within each level of a confounder, retraining at every draw, and read the displacement of the resulting AUC distribution from 0.5 as evidence of confounding; the conditional permutation test of Berrett et al. (2020) permutes within the levels of a confounder for the same reason. Song & Lee (2026) find that a pooled frame-level AUROC in video anomaly detection mostly measures video-level ranking; Bulut (2026) and Chen et al. (2026) find the same confound in LLM reasoning-trajectory probes and judge gates; and headline numbers have been re-examined for properties of the metric rather than of the system (Schaeffer et al., 2023; Card et al., 2020; Rodriguez et al., 2021). These audits all re-read data or scores they hold.

Identification without the scores. Our setting begins after publication, when a reader holds only the reported figure and the benchmark's annotations, and the answer is a set rather than a value. The uncertainty is not sampling error, which for aggregate benchmark metrics is treated by Longjohn et al. (2024); our bounds are deterministic given the figure. Partial identification is standard in econometrics (Manski, 2003; Tamer, 2010), usually for a treatment effect; here the target is a published statistic's conditional permutation mean. Our contribution is to formulate and solve that

problem: the annotation-only extrema (Proposition 1), the attainment result for this side constraint (Theorem 1), and the constructions and cases that follow. Technically the problem reduces to a rank-constrained optimisation over the permutohedron (Geoffrion, 1974; Ziegler, 1995), which we solve constructively; the contribution is the formulation, the sharp characterisation and the audit it supports rather than the machinery.

3 PROBLEM SETUP: KNOWN INFORMATION, UNKNOWN RANKING, TWO TARGETS

The two targets of Q1 and Q2 are defined here, with the information a reader is assumed to hold. Let examples carry outcome $y \in \{0, 1\}$, score s and stratum b. Two levels of released information are considered: $\mathcal{I}_0 = (y, b)$, the annotations, and $\mathcal{I}_1 = (y, b, A_{\text{reported}})$, the annotations with the printed pooled AUC. The unknown is not the scores' values but the pooled ranking R they induce, and every ranking is admitted: nothing is assumed about the model or its training, so a bound here says what the release alone determines and a witness says what it fails to exclude. Because no restriction is placed on the unknown ranking, every bound below depends on a stratum only through its counts $\{(n_{+,b}, n_{-,b})\}$ and not on which particular examples carry the labels, so a paper that reports those counts with its pooled AUC has released enough, without a per-example table.

symbol	meaning
b, B	stratum label; number of strata
$n_{+,b}$, $n_{-,b}$	positive and negative examples in stratum b; n_+, n_- pooled
λ	share of success/failure pairs that fall inside a stratum
A_{pooled}	AUC over all $n_+ n_-$ pairs — the reported statistic
A_{within}	AUC over the same-stratum pairs only
A_{cross}	AUC over the different-stratum pairs only
$\bar{R}_b$	mean rank of stratum b's scores in the pooled ordering
ν	$\mathbb{E}[A_{\text{pooled}}]$, scores fixed, outcomes permuted within strata
n, R_i	items in total; the pooled rank of item i
p_b, w_i	stratum b's share of positives; the weight $p_{b(i)}$ that (3) gives item i
S	$\sum_i y_i R_i$, the positives' rank sum; an exact AUC fixes it, a rounded one confines it

Write the AUC in Mann–Whitney form (Mann & Whitney, 1947) over the $n_+ n_-$ success/failure pairs, ties at 0.5, and partition the pairs by whether the two examples share a stratum:

$$A_{\text{pooled}} \;=\; \lambda\, A_{\text{within}} \;+\; (1-\lambda)\, A_{\text{cross}}, \qquad \lambda \;=\; \frac{\sum_b n_{+,b}\, n_{-,b}}{n_+\, n_-}. \tag{1}$$

A_{within} is the concordance rate over the same-stratum pairs and A_{cross} over the rest; when either pair set is empty ($\lambda = 0$ or $\lambda = 1$) its term is omitted and (1) reduces to the other. The weight λ is fixed by the design and the outcomes, not by the score, and Appendix C computes it for each design audited here. It is a share of pairs, not a share of information.

Now permute y inside each stratum with the scores held fixed. This preserves each stratum's outcome count, the score vector and the assignment to strata, and under the permutation distribution the within-stratum association between label and score is removed. The ranks are fixed, and a permuted label y_i^π equals 1 with probability $p_{b(i)}$, the stratum's share of positives, so $\mathbb{E}_\pi[\sum_i y_i^\pi R_i] = \sum_i p_{b(i)} R_i = \sum_b n_{+,b} \bar{R}_b$ and the mean is available in closed form,

$$\nu(R;\, y, b) \;:=\; \mathbb{E}_\pi[A_{\text{pooled}}(R, y^\pi)] = \frac{\sum_b n_{+,b}\, \bar{R}_b - n_+(n_+ + 1)/2}{n_+ n_-}, \tag{2}$$

with $\bar{R}_b$ taken as midranks so that (2) and (1) agree under ties. We write ν when the arguments are clear (the appendices write $\mathbb{E}[A_{\text{pooled}} \mid H_0]$ where the permutation distribution is read as a null hypothesis). Intuitively ν is high when the strata holding more positives sit higher in the global ranking. It depends on the score only through where each stratum sits, weighted by how many positives it holds, so two rules on the same data can have different ν at the same λ. Equation (2) is a

descriptive reference for the evaluation as observed; reading the full permutation distribution as a test additionally requires exchangeability (Appendix D).

Both the reported statistic and ν are linear in the ranking. Let R range over the permutations of $\{1, \dots, n\}$, a higher score receiving a higher rank, and assume $n_+ n_- > 0$. Writing R_i for the pooled rank of item i and $w_i = p_{b(i)} = n_{+,b(i)}/(n_{+,b(i)} + n_{-,b(i)})$ for its stratum's share of positives,

$$A_{\text{pooled}} = \frac{\sum_{i:\, y_i=1} R_i - n_+(n_+ + 1)/2}{n_+ n_-}, \qquad \nu = \frac{\sum_i w_i R_i - n_+(n_+ + 1)/2}{n_+ n_-}. \tag{3}$$

The actual AUC weights each rank by the item's own label; the permutation mean weights the same rank by the item's stratum positive rate. An exact pooled AUC therefore fixes $y^\top R = S$, the sum of the positive examples' ranks. That is one linear constraint on R, and it does not in general determine $w^\top R$. That gap is the identification problem of Section 4.

Identity (1) also says what ν is not. Permuting the labels inside a stratum disturbs more than that stratum's own pairs: an example that switches from positive to negative also switches sides in every comparison it makes with the other strata. Subtracting the permutation mean from the pooled figure gives $A_{\text{pooled}} - \nu = \lambda(A_{\text{within}} - \frac{1}{2}) + (1 - \lambda)(A_{\text{cross}} - \mathbb{E}_\pi A_{\text{cross}})$. The permutation moves the cross term as well, so the difference carries that term's own displacement.

Read ν, then, only as a reference level for the pooled statistic. Not even the sign of $A_{\text{pooled}} - \nu$ fixes the sign of $A_{\text{within}} - \frac{1}{2}$: in the eight-item design of Section 4 one ordering has $A_{\text{pooled}} = 0.5625$, $\nu = 0.5$ and $A_{\text{within}} = 1/3$. ν says where the pooled AUC's mean lies under within-stratum randomisation, and no more: the observed labelling is itself one permutation among many, so ν alone does not say how extreme the observed AUC is; that needs the spread as well as the mean.

One caution follows from this and runs through everything below. Excluding $\nu = 0.5$ establishes neither within-stratum discrimination, significance nor deployment gain; and even a within-stratum AUC, measured rather than bounded, answers only which pairs are compared, not whether the score reads their content (Section 5.2). Section 5 takes up what ν leaves unidentified.

4 Identification results

This section answers Q1 first — what the release fixes about the reference level ν — and then Q2, what it fixes about within-stratum discrimination.

Annotations alone. Equation (2) needs the scores only through $\bar{R}_b$, and varying the ranking over all rankings compatible with $\mathcal{I}_0$ gives a minimum and a maximum computable without any score. Because strata differ in their share of positives, the rank positions a score gives each stratum's members already move ν.

Proposition 1. Fix the design — the strata and their outcome counts. Over all scores, ν is maximised when the strata occupy contiguous rank blocks with p_b non-decreasing in rank, and minimised when p_b is non-increasing. Both extremes are computable from the outcomes and stratum membership alone.

Proof. Only the numerator $\sum_b n_{+,b}\bar{R}_b$ of (2) depends on the score. Take any assignment of items to ranks and any two items $i \in a$, $j \in b$ with $a \neq b$ and $r_i < r_j$. Exchanging their ranks changes $\bar{R}_a$ by $(r_j - r_i)/n_a$ and $\bar{R}_b$ by $(r_i - r_j)/n_b$, so it changes the numerator by $(r_j - r_i)\,(p_a - p_b)$. At a maximiser no exchange increases the numerator, so $p_a \leq p_b$ for every such pair: p is non-decreasing along the ranking, so a maximiser can be taken to be a contiguous-block arrangement sorted by p_b, strata of equal p_b being free to interleave. Reversing the inequality gives the minimiser. □

Reversing a score sends ν to one minus itself, so the extremes are symmetric about 0.5 and the interval always contains it, and its endpoints are attained.

Adding the reported AUC. Take eight examples in two strata of four, three correct in the first and one in the second. Over every ordering a score could impose, ν ranges over $[0.25, 0.75]$, Proposition 1's interval. A reported pooled AUC pins down one thing about the ordering, the sum of the ranks assigned to the four positive examples: fixing the pooled AUC at 0.875 leaves ν in $[0.46875, 0.75]$, and fixing it at 1.000 leaves $[0.53125, 0.71875]$, which no longer contains 0.5.

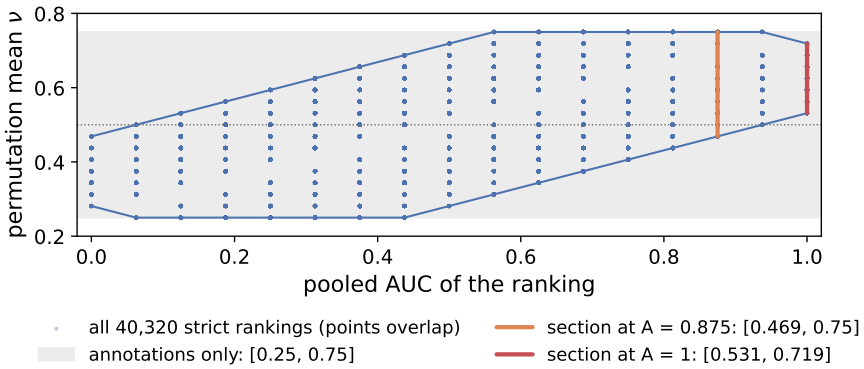

Figure 1: The eight-item design: all 40,320 strict rankings as points (pooled AUC, ν), overlapping heavily. Lines: the smallest and largest ν at each attainable AUC. The highlighted segments are the interval hulls at $A = 0.875$ and $A = 1$ (not every point inside is attained); the grey band is the annotation-only interval of Proposition 1.

Within-stratum performance is constrained here too: at 0.875 the within-stratum AUC can only be $2/3$, $5/6$ or 1 among strict rankings, with ties adding just $3/4$ and $11/12$, so one half and below is ruled out. Realistic designs need not behave this way (Section 5.2).

Figure 1 shows all $8! = 40{,}320$ orderings as points: the identified set at a reported AUC is a vertical section of that cloud, and the bounds are its ends. For realistic designs exhaustive enumeration is infeasible, and the rest of this section derives the bounds directly.

Why the bound is attained. Fixing the AUC fixes the sum S of the ranks the positives hold. We want the smallest stratum-weighted rank sum $w^\top R$ among rankings with that positive rank sum: an assignment problem with one side constraint. A multiplier θ on the constraint turns it into a sorting problem, since for fixed θ the relaxed objective $\sum_i (w_i - \theta y_i) R_i$ is minimised by sorting the adjusted weights. Raising θ makes the positives cheaper to place high, so the sorted solution's positive rank sum climbs, in steps, from the smallest value any ranking gives to the largest. At a breakpoint two adjusted weights tie, and every arrangement of a tied class is equally optimal, so the relaxed minimisers there span a contiguous interval of positive rank sums. At the optimal multiplier that interval contains the required S, so rearranging inside the tied class yields a feasible ranking with the same relaxed objective, and its objective is the bound (Appendix A gives the integer-connectivity step in full).

A dual bound. Relaxing the constraint with a multiplier gives an immediate bound. With R ranging over the permutations of $\{1, \dots, n\}$, for every θ,

$$\min_R \Big\{ \textstyle\sum_i w_i R_i \;:\; \sum_i y_i R_i = S \Big\} \;\geq\; \min_R \sum_i (w_i - \theta y_i) R_i \;+\; \theta S \;=:\; D(\theta), \tag{4}$$

and symmetrically $\max_R \{ w^\top R : y^\top R = S \} \leq \max_R \sum_i (w_i - \theta y_i) R_i + \theta S =: \bar{D}(\theta)$.

Theorem 1 (attainment)**.** Let $S_{\min} = n_+(n_+ + 1)/2$ and $S_{\max} = n_+ n - n_+(n_+ - 1)/2$ be the positives' rank sums when they occupy the lowest and the highest ranks. For strict rankings and every integer $S \in [S_{\min}, S_{\max}]$, $\max_\theta D(\theta)$ equals the constrained minimum of $w^\top R$ and $\min_\theta \bar{D}(\theta)$ the constrained maximum, and a ranking attaining each is computable. Both bounds are therefore sharp: the identified set for ν at that S lies in the interval and contains its endpoints. In the units of (3) those endpoints are $(\max_\theta D(\theta) - n_+(n_+ + 1)/2)/(n_+ n_-)$ and $(\min_\theta \bar{D}(\theta) - n_+(n_+ + 1)/2)/(n_+ n_-)$.

Proof sketch. D is concave and piecewise linear, and at a maximiser θ^* zero lies in its superdifferential, which places the required S in the interval of rank sums the relaxed minimisers realise. One tied-class rearrangement then attains S exactly. Appendix A gives the argument in full, and the maximum. □

Computation. Each evaluation of D is one $O(n \log n)$ sort, and a one-dimensional search on the concave D locates θ^*. By (4) any θ gives a valid bound, so the search precision affects how far the printed endpoint lies from the attained one, never validity; the certificate is a feasible ranking paired with the dual value its objective matches, so optimality does not rest on the search. Appendix A gives

the search, the tolerances and the primal–dual gaps, and Appendix B relates Theorem 1 to known results.

Tied scores. The ties in the proof are ties in the adjusted weights c_i, not in the scores. A tied score induces midranks, whose positive rank sum can be a half-integer; a midrank vector is an average of permutations, so the dual bounds remain valid at half-integer S (Appendix A).

Corollary 1 (tied scores)**.** For every $S \in \mathbb{Z} + \frac{1}{2}$ with $S_{\min} < S < S_{\max}$, $\max_\theta D(\theta)$ is the minimum of $w^\top R$ over weak orderings with positive rank sum S, attained by a weak ordering with a single tied pair; likewise for the maximum; so the bounds are sharp at every attainable rank sum, strict or tied (proof in Appendix A).

Printed precision. A figure printed as r at three decimals is read as the closed band $[r - 0.0005, r + 0.0005]$, closed so that either usual rounding convention is covered and the printed bounds can only move outward. The band's real ends are in general not attainable rank sums, so the bounds are evaluated at the attainable rank sums nearest each end, where the constructions above attain them, and the dual at the real ends bounds them from outside. For every case reported the extreme falls at an end of the band. The tables therefore give valid bounds attained by explicit rankings, rounded outward to three decimals (Appendix A).

Q2: within-stratum performance. The same release also constrains the second target. We take the pair-weighted within term of (1), the estimand the decomposition defines (Appendix G discusses the unweighted alternative),

$$A_{\text{within}} \;=\; \frac{\sum_b n_{+,b}\, n_{-,b}\, A_b}{\sum_b n_{+,b}\, n_{-,b}}, \tag{5}$$

with A_b the AUC inside stratum b and permutation reference 0.5 by construction. We do not attempt to derive the full identified set for A_{within}. We characterise exactly when its two endpoints remain feasible, and construct low-within witnesses, which is enough to decide whether qualitatively different readings of a reported figure remain possible — the question an auditor asks first. A reported figure settles exactly whether the two endpoints $A_{\text{within}} = 1$ and $A_{\text{within}} = 0$ are feasible; which intermediate values it admits is not settled here.

Fixing $A_{\text{within}} = 1$ pins the within-stratum term of (1) at its maximum, so the only freedom left is how the strata are ordered against one another — and that is what Proposition 1 already extremised. The shift is $\lambda/2$: under within-stratum permutation the within term sits at $\frac{1}{2}$, and forcing $A_{\text{within}} = 1$ raises it to 1, so by (1) the pooled figure gains $\lambda(1 - \frac{1}{2})$. The threshold below is therefore the annotation-only endpoint plus that amount.

Proposition 2 (within-stratum extremes)**.** Assume $\lambda > 0$ and let $[\nu_{\min}, \nu_{\max}]$ be the annotation-only interval of Proposition 1. Over rankings with $A_{\text{within}} = 1$ (every positive above every negative of its own stratum), the pooled AUC takes exactly the attainable values in $[\nu_{\min} + \lambda/2,\ 1]$; over rankings with $A_{\text{within}} = 0$, exactly those in $[0,\ \nu_{\max} - \lambda/2]$. A reported figure is consistent with $A_{\text{within}} = 1$ iff its rounding band contains an attainable value of the first set, and with $A_{\text{within}} = 0$ iff it contains one of the second; the two witnesses may have different exact AUCs inside the band. Proposition 3 exhibits some of the intermediate values.

Proof sketch. Placing all positives above all negatives meets the constraint and gives pooled AUC 1. The smallest value comes from a block argument: under $A_{\text{within}} = 1$ an optimum can be taken with each stratum contiguous, its negatives below its positives, and the blocks ordered by non-increasing p_b; that arrangement's cross-stratum concordance equals its permutation mean, so its pooled AUC is $\nu_{\min} + \lambda/2$. Every integer rank sum between is reached by swapping a negative that sits directly above a positive of another stratum, and tying such a pair reaches the half-integers; Appendix A shows that the feasible orderings are connected by such adjacent swaps. The $A_{\text{within}} = 0$ case is the mirror image. Appendix A proves the contiguity step, relates it to Lawler (1978)'s scheduling rule and gives the exhaustive checks. □ The two conditions are separate; Appendix A gives a design where a printed value admits both extremes although no exact AUC does.

Intermediate values. Inverting a chosen set of strata and leaving the rest concordant fixes the within-stratum AUC at a known value strictly between 0 and 1, and pins the exact range of pooled AUCs compatible with it.

Proposition 3 (partly inverted arrangements)**.** Let I be any set of mixed strata and $\lambda_I = \sum_{b\in I} n_{+,b}n_{-,b}/(n_+n_-)$. Over rankings in which every stratum of I is fully inverted (its negatives above its positives) and every other stratum fully concordant, $A_{\text{within}} = 1 - \lambda_I/\lambda$, and the pooled AUC takes exactly the attainable values in an interval $[L(I), U(I)]$ whose ends are attained by explicit block arrangements. A reported figure whose band contains an attainable value of that interval is therefore consistent with $A_{\text{within}} = 1 - \lambda_I/\lambda$; Proposition 2 is the case of I empty and of I all strata.

Proof. Appendix A; the two endpoints come from the run-exchange argument of Proposition 2, and the feasible arrangements are the linear extensions of one partial order, joined by adjacent transpositions that move the rank sum by at most one. □

Section 5.2 uses these only to exhibit witnesses below one half.

5 Empirical analysis

5.1 Q1 in practice: how far the reported figure constrains the reference level

What does the reported AUC add? Across six public designs the annotation-only interval is wide and, by symmetry, contains 0.5, from ProcessBench MATH's $[0.066, 0.934]$ to MMLU's (Hendrycks et al., 2021) $[0.292, 0.708]$, with λ below 0.02 on every one (Appendix C). Table 1 applies Section 4 to seven scoring rules on OpenAI's released PRM800K samples, conditioning on each figure as a paper would print it: four aggregations of the per-step ratings, the first step's rating alone, the released solution-level score and an outcome-reward model (ORM) comparator. Its last column reports a witness. A score built from the labels is not a claim that any model behaves that way: it shows that if two score vectors both match the released information and differ in within-problem AUC, that information does not separate them. Six of the seven rules have a pooled AUC above 0.89, and for each the sharp bounds exclude 0.5: the ORM's 0.941 confines ν to $[0.562, 0.908]$ (extremes rounded outward). The exception is the rule printed at 0.617, whose bounds $[0.294, 0.838]$ still contain 0.5. Figure 2 sweeps the printed value: the lower bound crosses 0.5 near 0.874, above which a reported figure narrows the interval from below.

Does sharpness matter empirically? A naive bound, the *unconstrained rank-difference bound*, ranges the rank difference of (3) over all rankings without requiring the same ranking to produce the printed figure: valid, but not necessarily sharp (Appendix A). At the high reported values it reaches the same verdict on 0.5 as the sharp bounds, so there the side constraint buys no change of conclusion. At lower reported values the constraint can tighten the interval substantially: for the first-step rule the naive $[0.236, 0.910]$ becomes $[0.294, 0.838]$. Appendix Table 3 reports the sweep over four designs. The main gain is therefore structural rather than uniformly numerical: no narrower interval contains every ν compatible with $\mathcal{I}_1$, because an explicit ranking that reproduces the reported figure attains each endpoint. The uncertainty that remains is the released information's, not the solver's.

Do the bounds contain the real ν? Where scores are released, ν can be measured and the bounds checked against it by withholding the scores. Table 1 does this for the seven PRM800K rules: their ν run from 0.607 to 0.797, the lowest of them 0.107 above 0.5, and each lies inside its row's bounds. The first-step rule shows what reading against 0.5 misses: its pooled AUC is 0.617 but its ν is already 0.607, and its within-problem AUC 0.526. Appendix D gives that figure's bootstrap interval and the rule's best-of-N selection gain. The same check on LM-Polygraph's 35 MMLU methods is in Appendix F.

5.2 Q2 in practice: which within-stratum values remain compatible

Applying Propositions 2–3 to the audited designs: on PRM800K, $A_{\text{within}} = 1$ needs a printed value of at least 0.091 and $A_{\text{within}} = 0$ one of at most 0.909: the first-step and mean rules rule out neither extreme, while the five figures printed at 0.919 or above are inconsistent with $A_{\text{within}} = 0$. On ProcessBench the thresholds are 0.067 and 0.933 (`within_extremes.py`).

Inverting a set I of strata as above and searching nested families of such sets (`within_family.py`) gives, for each PRM800K figure, the smallest within-problem AUC found compatible with it in those families (last column of Table 1). Within-problem AUCs below the constructed value are uncertified

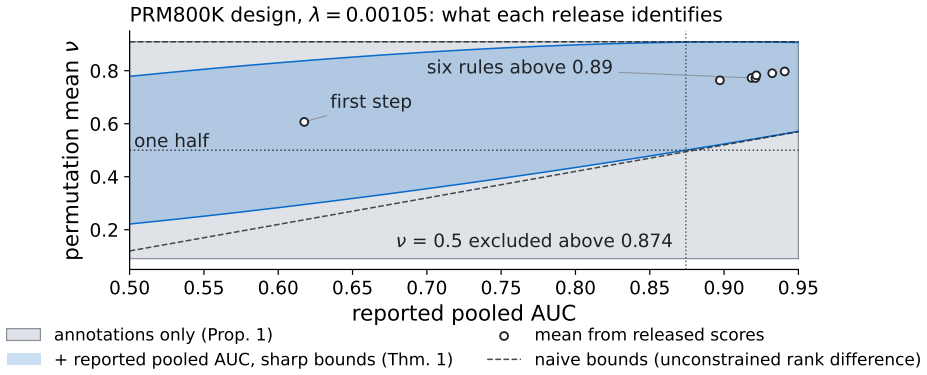

Figure 2: Bounds on the conditional permutation mean ν as a function of the reported pooled AUC, PRM800K design ($\lambda = 0.00105$). Grey: annotations only (Proposition 1), $[0.090, 0.910]$. Blue: sharp bounds given the reported figure (Theorem 1), the envelope of a discrete identified set rather than a confidence interval. Dashed: the naive bounds of Section 5.1, whose upper bound coincides with the annotation-only upper end throughout this range. Circles: ν measured from the released scores of the seven rules (Table 1). Dotted vertical line: the printed value, about 0.874, above which no score reproducing the figure has $\nu = 0.5$. Below 0.5 the picture is the image under $(A, \nu) \mapsto (1-A, 1-\nu)$.

rather than excluded. Every one of the seven reported pooled AUCs exceeds $\nu_{\min} + \lambda/2$, so each admits $A_{\text{within}} = 1$; each also admits a within-problem AUC below one half, though five rule out exactly zero (on MMLU, the top method's 0.793 admits 0.184).

For any pair of these reported rules the pooled ordering need not be the within-problem ordering: the lower figure can be paired with a witness of within-problem AUC 1 and the higher with a witness below one half, so the order the pooled figures induce need not be the order within problems. Such a reversal needs a pair of witnesses, and at an exact pooled AUC of 1 no low-within witness exists: there every positive outranks every negative, so the within-problem AUC is 1 as well.

A published figure with unreleased scores. A process-reward paper (Lee et al., 2026) reports a step-level ROC-AUC of 0.765 over the 1,000 solutions of ProcessBench's MATH split, describing it as ranking correct steps above incorrect ones. The released code fixes the pooling and the per-step outcome, which follows from ProcessBench's (Zheng et al., 2025) public first-error index; the scores are not. We read the published value as a standard Mann–Whitney AUC; the released implementation departs from that definition, and we could not check which routine produced 0.765 (Appendix E). The constructions below reproduce the published value under both.

Table 1: Seven scoring rules on PRM800K: 815,632 solutions to 500 problems, $n_+ = 433{,}067$, $n_- = 382{,}565$, $\lambda = 0.00105$ throughout. "pooled" is our recomputation from the release, and the bounds condition on its three-decimal rounding: "naive" is the unconstrained rank-difference bound, "sharp" those of Theorem 1, both rounded outward. Writing A_{w} for A_{within}: ν and A_{w} obs. are measured from the released scores, while "witness A_{w}" is the lowest within-problem AUC of an alternative score we build that matches the same printed figure: the lowest such score found among the families searched in Section 5.2. It bounds the smallest feasible value from above and is not a sharp lower bound.

rule	pooled	naive	sharp	excl. ν=0.5	ν	A_{w} obs.	witness A_{w}
ORM	0.94078	[0.560, 0.910]	[0.562, 0.908]	yes	0.797	0.884	0.175
released	0.93238	[0.551, 0.910]	[0.553, 0.909]	yes	0.791	0.881	0.118
product	0.92170	[0.541, 0.910]	[0.544, 0.909]	yes	0.783	0.871	0.057
minimum	0.92110	[0.540, 0.910]	[0.543, 0.910]	yes	0.772	0.868	0.052
last	0.91853	[0.538, 0.910]	[0.541, 0.910]	yes	0.773	0.867	0.042
mean	0.89718	[0.516, 0.910]	[0.520, 0.910]	yes	0.764	0.844	0
first	0.61747	[0.236, 0.910]	[0.294, 0.838]	no	0.607	0.526	0

The bounds of Section 4 give $\nu \in [0.434, 0.901]$, so the figure does not exclude 0.5, and $\lambda = 0.00054$. Whether within-solution discrimination is constrained is settled by Proposition 2, not by those bounds: 0.765 lies inside $[0.067, 0.933]$, so within-solution AUC 0 and 1 are both feasible at that precision. The first row of Table 2 inverts the correct/incorrect order in all 479 solutions holding both. Two score families match all four labelling variants the paper compares at printed precision, their within-solution AUCs following the pooled-AUC ordering in one family and reversing it in the other (Appendix E).

Table 2: ProcessBench MATH: rows 1 and 3 match 0.765 at printed precision, with within-solution AUC 0 and 1.

score	pooled AUC	ν	within-solution AUC	uses labels
inverting witness	0.765	0.900	0.000	yes
position only	0.715	0.531	1.000	no
position plus per-solution offsets	0.765	0.765	1.000	yes

A within-solution AUC restricts comparisons to pairs drawn from the same solution; a pooled AUC does not. Even $A_{\text{within}} = 1$ need not imply content sensitivity: Appendix E gives a position-only score that attains it without locating the first error.

6 Discussion and reporting implications

When a pooled AUC is the right statistic. If the decision the score serves ranks candidates across strata — a global triage queue, a corpus-wide filter — then the pooled figure is the estimand, and nothing here argues against reporting it. The mismatch matters when the deployed comparison is inside a stratum: choosing among candidate answers to one problem, ranking the steps of one solution, abstaining on one passage. There the pooled figure can be high while the within-stratum comparison the deployment actually makes is uninformative, and Section 5.2 exhibits scores where exactly that holds.

What to report. The counts with the reported figure already give the bounds of Section 4. Different releases answer different questions. Counts with the pooled figure give the Q1 bounds; adding each stratum's mean rank $\bar{R}_b$, or ν alone, answers Q1 exactly; each stratum's own AUC A_b answers Q2; a per-example table of scores, outcomes and stratum identifiers answers both. The lightest release that lets a reader reproduce the bounds is the per-stratum count table beside the pooled figure: far smaller than a per-example release, it is what our bounds consume, and a paper that prints it alongside its pooled figure lets any reader run the audit of Section 4 without the scores. Appendix I reports how we searched for published figures this applies to, and what the search returned.

Three limitations qualify these results. (i) Evaluator: the ProcessBench bounds assume the reported number is a standard Mann–Whitney AUC. Our witnesses reproduce the published value under both the standard and the released routine, but that does not establish which routine produced it (Section 5.2). (ii) Stratification: ν depends on the scores, the outcomes and the stratification together, so the choice must follow the decision the score serves; Appendix H varies the stratification of one fixed evaluation and reports what moves. (iii) Interpretation: the results say what the release determines, not what the score actually did or what its authors intended. The interval is the hull of the identified set: its endpoints are attained, but not every interior value need be.

7 Conclusion

Identifying a statistic and measuring the ability a reader cares about are different questions. From labels, stratum memberships and a reported pooled AUC a reader can bound that figure's conditional permutation mean sharply, and decide which within-stratum extremes it rules out. The release can constrain Q1 substantially while leaving Q2 weakly identified: on PRM800K the five figures at 0.919 and above rule out a within-problem AUC of exactly zero, yet all seven stay compatible with one below a half. A pooled AUC alone should not carry a within-stratum claim, and Section 6 says what to publish beside it.

REPRODUCIBILITY STATEMENT

The scripts are released as a supplementary archive whose single entry point is `verify_artifact.py`, which checks every number in this paper against the released result files and prints a pass/fail line for each; the scripts named below regenerate those files from the inputs. Four checks play four different roles and none substitutes for another: the proofs of Section 4 establish the bounds and their attainment for strict rankings at an exact AUC; the exhaustive enumerations of Appendix A, implemented independently of the solver, check the solver on every small design; the input hashes below fix which files were read, though not that we parsed them correctly; and the primal–dual certificate of Table 4 shows, for each reported endpoint, a permutation with the required rank sum whose objective equals the dual value. Regeneration by itself shows only that the numbers come from the code.

Inputs. Section 5 derives from public releases; the script reports the byte count and SHA-256 of each file it reads. Appendix E's case reads ProcessBench's MATH split from its published parquet and the reported figures from the paper's Table 3, and `processbench_case_study.py` rebuilds the design, the annotation-only interval and the certified regions from them; `processbench_ranking_reversal.py` builds the position-only score and the two families that match all four published figures. `survey_manual_readjudication.json` holds the eight coded fields and quotes for each of the 34 papers the screen flags. `exact_region.py` locates θ^* and builds the ranking attaining each endpoint of Table 1, with `band_slopes.py`, `certificate_gap_sweep.py`, `certificate_sweep.py` and `make_fig_eight.py` covering the band-end sign check, the comparison against exhaustive enumeration, Figure 2's band and the eight-item enumeration of Figure 1, `ties_corollary_check.py` runs the construction of Corollary 1, `eight_item_tied_within.py` enumerates the eight-item design's weak orderings, `within1_contiguity.py` checks the contiguity lemma behind Proposition 2, and `mmlu_within_ci.py` and `first_step_within_ci.py` the cluster-bootstrap intervals quoted in Section 5. Appendix H's uncertainty scores derive from public QA datasets under a protocol fixed in writing before the scores existed. Section 5's three constructions are synthetic and self-contained, and Proposition 1's exhaustive check enumerates every ordering on 139 generated designs.

Everything reproduces on a laptop (Apple M1 Pro, 32 GB, no GPU; NumPy 2.0). The largest input is a 2.1 GB released file read in one streaming pass. The endpoint solver is repeated sorts over the 815,632 PRM800K items: one rule's two endpoints, each evaluated at three points of the band and re-certified, take about two minutes in total (`exact_region.py`), and the 50-value sweep of Appendix A takes 36 minutes on PRM800K and under half a minute on the other three designs together. The 2,000 within-problem permutations per rule are a separate precomputation over the same items. Bootstrap counts are 2,000 over strata, and 4,000 for the intervals quoted in Section 5. `verify_artifact.py` itself, which only reads the result files, runs in about a second.

ETHICS STATEMENT

This work re-analyses public releases. No human subjects, no personal data and no model training: Appendix H scores one released instruction-tuned model's generations on public QA data, and its six detectors are computed from those generations rather than fitted. Section 5 re-analyses the released scores of the PRM800K and LM-Polygraph artifacts to characterise a reporting convention, and the statistic those two artifacts report is not the one we recompute. Appendix I codes published papers by textual pattern; we report aggregate counts and our own coding's error rate, name no paper as mistaken, and release both the current and the superseded coding. Appendix E names one paper, whose reported figures we take as published and do not dispute; what we compute there is what its released information leaves a reader able to say, and the case is legible only because that paper released its evaluation code.

AI USE STATEMENT

An AI assistant was used throughout this work: to search for and check related work, to write and run the analysis scripts, to generate the randomised synthetic designs used for verification, to draft and

edit prose, and to draft some of the proof arguments, including the adjacent-exchange argument in the proof of Proposition 2. The authors reviewed every statement and proof and are responsible for all of them.

Three checks are independent of that assistance. `verify_artifact.py` recomputes every number reported here from the released result files; the exhaustive enumerations of Appendix A test the propositions against brute force on small designs; and the input hashes fix which files were read. The first guards the numbers and not the mathematics, the second the mathematics on small cases only, and none certifies that we read the inputs' labels and strata correctly. The research questions, the framing and the final content are the authors'.

REFERENCES

S. Abdulaziz and E. Bondarev. Auditing frame-level AUC in weakly supervised video anomaly detection: granularity, resolution, and scene bias. In *ECCV 2026 Empirical Theory Workshop*, 2026. arXiv:2608.11985.

S. Arkhangelskiy. PhAIL: a real-robot VLA benchmark and distributional methodology. *arXiv preprint arXiv:2605.29710*, 2026.

D. Bamber. The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *Journal of Mathematical Psychology*, 12, 1975.

T. B. Berrett, Y. Wang, R. F. Barber, and R. J. Samworth. The conditional permutation test for independence while controlling for confounders. *Journal of the Royal Statistical Society: Series B*, 82(1):175–197, 2020.

D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of the 2019 World Wide Web Conference*, pp. 491–500, 2019.

Y. U. Bulut. It's the problem, not the path: budget and difficulty confounds in LLM reasoning trajectories. *arXiv preprint arXiv:2609.03436*, 2026.

D. Card, P. Henderson, U. Khandelwal, R. Jia, K. Mahowald, and D. Jurafsky. With little power comes great responsibility. In *EMNLP*, pp. 9263–9274, 2020.

E. Chaibub Neto, A. Pratap, T. M. Perumal, M. Tummalacherla, B. M. Bot, L. Mangravite, and L. Omberg. A permutation approach to assess confounding in machine learning applications for digital health. In *KDD*, pp. 54–64, 2019.

C. Chen et al. Competence, not accuracy: a diagnostic for reference-free judge gates in skill optimization. *arXiv preprint arXiv:2608.18719*, 2026.

A. Cherif. HalluScan: a systematic benchmark for detecting and mitigating hallucinations in instruction-following LLMs. *arXiv preprint arXiv:2605.02443*, 2026.

E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44(3), 1988.

J. Duan, H. Cheng, S. Wang, A. Zavalny, C. Wang, R. Xu, B. Kailkhura, and K. Xu. Shifting attention to relevance: towards the predictive uncertainty quantification of free-form large language models. In *ACL*, pp. 5050–5063, 2024.

A. M. Geoffrion. Lagrangean relaxation for integer programming. *Mathematical Programming Study*, 2:82–114, 1974.

S. Gururangan et al. Annotation artifacts in natural language inference data. In *NAACL*, 2018.

D. J. Hand. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning*, 77(1), 2009.

G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, 1934.

F. E. Harrell, K. L. Lee, and D. B. Mark. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine*, 15(4):361–387, 1996.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *ICLR*, 2021.

V. Inácio and M. X. Rodríguez-Álvarez. The covariate-adjusted ROC curve: the concept and its importance, review of inferential methods, and a new Bayesian estimator. *Statistical Science*, 37 (4), 2022.

H. Janes and M. S. Pepe. Adjusting for covariate effects on classification accuracy using the covariate-adjusted receiver operating characteristic curve. *Biometrika*, 96(2), 2009.

T. Jiang et al. What are we actually benchmarking in robot manipulation? *arXiv preprint arXiv:2606.04233*, 2026.

N. Kallus and A. Zhou. The fairness of risk scores beyond classification: bipartite ranking and the xAUC metric. In *NeurIPS*, 2019. arXiv:1902.05826.

E. L. Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals of Discrete Mathematics*, 2:75–90, 1978.

N. Lee, S. Hong, and J. Lee. Efficient process reward modeling via contrastive mutual information. In *ACL*, 2026. arXiv:2604.10660.

H. Lightman et al. Let's verify step by step. In *ICLR*, 2024. arXiv:2305.20050.

R. Longjohn, G. Gopalan, and E. Casleton. Statistical uncertainty quantification for aggregate performance metrics in machine learning benchmarks. In *NeurIPS 2024 Workshop on Statistical Frontiers in LLMs and Foundation Models*, 2024. arXiv:2501.04234.

P. Manakul, A. Liusie, and M. J. F. Gales. SelfCheckGPT: zero-resource black-box hallucination detection for generative large language models. In *EMNLP*, 2023.

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

C. F. Manski. *Partial Identification of Probability Distributions*. Springer, 2003.

N. A. Obuchowski. Nonparametric analysis of clustered ROC curve data. *Biometrics*, 53(2), 1997.

A. Poliak et al. Hypothesis only baselines in natural language inference. In **SEM*, 2018.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

S. Reddy, D. Chen, and C. D. Manning. CoQA: a conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

J. Ren, J. Luo, Y. Zhao, K. Krishna, M. Saleh, B. Lakshminarayanan, and P. J. Liu. Out-of-distribution detection and selective generation for conditional language models. In *ICLR*, 2023. arXiv:2209.15558.

P. Rodriguez, J. Barrow, A. Hoyle, J. P. Lalor, R. Jia, and J. Boyd-Graber. Evaluation examples are not equally informative: how should that change NLP leaderboards? In *ACL-IJCNLP*, pp. 4486–4503, 2021.

R. Schaeffer, B. Miranda, and S. Koyejo. Are emergent abilities of large language models a mirage? In *NeurIPS*, 2023. arXiv:2304.15004.

E. H. Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B*, 13(2), 1951.

W. E. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3 (1–2):59–66, 1956.

I. Song and J. Lee. Frame-level evaluation in weakly supervised video anomaly detection mostly measures video-level ranking. *arXiv preprint arXiv:2608.21854*, 2026.

A. Sudjianto and A. J. Liu. Decomposing global AUC into cluster-level contributions for localized model diagnostics. *arXiv preprint arXiv:2508.07495*, 2025.

A. Sun and K. Yang. RouteGuard: certifying routing gain in LLM multi-agent systems when complementarity is not enough. *arXiv preprint arXiv:2608.07583*, 2026.

E. Tamer. Partial identification in econometrics. *Annual Review of Economics*, 2(1):167–195, 2010.

A. Tschalzev, S. Lüdtke, H. Stuckenschmidt, and C. Bartelt. Lost in aggregation: how benchmarks overlook irreplaceable model strengths. *arXiv preprint arXiv:2608.18919*, 2026.

P. van Elteren. On the combination of independent two-sample tests of Wilcoxon. *Bulletin of the International Statistical Institute*, 37:351–361, 1960.

H. C. van Houwelingen and H. Putter. *Dynamic Prediction in Clinical Survival Analysis*. CRC Press, 2011.

R. Vashurin, E. Fadeeva, A. Vazhentsev, et al. Benchmarking uncertainty quantification methods for large language models with LM-Polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248, 2025.

A. Vazhentsev, G. Kuzmin, A. Tsvigun, A. Panchenko, M. Panov, M. Burtsev, and A. Shelmanov. Hybrid uncertainty quantification for selective text classification in ambiguous tasks. In *ACL*, pp. 11659–11681, 2023.

P. Wang et al. Math-Shepherd: verify and reinforce LLMs step-by-step without human annotations. In *ACL*, 2024.

C. Zheng et al. ProcessBench: identifying process errors in mathematical reasoning. In *ACL*, 2025.

G. Zhou et al. Deep interest network for click-through rate prediction. In *KDD*, 2018.

G. M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer, 1995.

A PROOFS, CERTIFICATES AND CHECKS

Proposition 2, in full. Placing all positives above all negatives meets the constraint and gives pooled AUC 1. For the smallest value, list the items from rank 1 upward; the positives' rank sum is the sum of their positions, and the constraint says that each stratum's negatives come before its positives. Two facts settle the minimum. First, some optimal list keeps each stratum contiguous, negatives then positives: exchanging two adjacent runs of one stratum's items changes the rank sum by an amount whose sign orders the runs by their share of positives, and that order forces a split stratum's parts to be separated only by runs of equal share, which can therefore be exchanged at no cost until the parts meet. The bookkeeping is below, and the argument is the unit-time, 0/1-weight case of Lawler (1978)'s rule for series–parallel precedence constraints. Second, with blocks contiguous, exchanging adjacent blocks a (below) and b changes the positives' rank sum by $n_{+,a}n_b - n_{+,b}n_a = n_a n_b(p_a - p_b)$, which is negative exactly when $p_a < p_b$, so the optimum has p_b non-increasing in rank. Cross-stratum concordance under contiguous blocks depends only on the block order: a cross-label pair drawn from two different strata is concordant exactly when the positive's block sits above the negative's, and permuting labels within strata changes neither the block order nor any $(n_{+,b}, n_{-,b})$, so A_{cross} is constant over the permutation distribution and equals its own mean. Only the within term moves, from $A_{\text{within}} = 1$ to its permutation mean $1/2$, so by (1) this ranking has $A_{\text{pooled}} = \lambda + (1-\lambda)A_{\text{cross}}$ against $\nu = \lambda/2 + (1-\lambda)A_{\text{cross}}$, that is $A_{\text{pooled}} = \nu + \lambda/2$, which at this block order is $\nu_{\min} + \lambda/2$ by (2) (checked exhaustively on small designs below). In any feasible ranking with pooled AUC below 1 some negative sits directly above a positive; they belong to different strata, and swapping them raises the rank sum by one while keeping the constraint, so every integer rank sum between is attained. Tying such a pair instead of swapping it attains the half-integer between, so with tied scores every attainable rank sum is reached. The $A_{\text{within}} = 0$ case is the mirror image. □ The conditions are separate because on a design of 2,000 two-item strata a printed 0.500 admits both extremes although no exact AUC does.

Corollary 1, proof. At a maximiser θ^* the superdifferential contains 0, so $S^-(\theta^*) \le S - \frac{1}{2}$ and $S + \frac{1}{2} \le S^+(\theta^*)$, both integers. Within one tie class of the adjusted weights c_i the relaxed objective is unchanged by any rearrangement of that class's ranks, while swapping a positive and a negative that hold adjacent ranks inside it moves $y^\top R$ by exactly one; such swaps connect the class's orderings, and concatenating the paths across classes gives a relaxed minimiser at every integer rank sum between $S^-(\theta^*)$ and $S^+(\theta^*)$, consecutive sums one swap apart. Run at the two rank sums above, the block construction therefore returns solutions R' and R'' differing only by swapping one positive and one negative holding adjacent ranks. Giving those two items equal scores yields the midrank vector $(R' + R'')/2$, which has positive rank sum S and, both objectives being linear, the same objective $D(\theta^*)$. □

Proposition 3, partly inverted arrangements. Writing P_b, N_b for stratum b's positives and negatives and reading each arrangement from the bottom rank upward (P_I, N_I for the positives and negatives of the inverted strata, P_C, N_C for those of the concordant ones), the two ends of $[L(I), U(I)]$ are

$$L(I): \quad P_I \;\middle|\; \big[N_b P_b\big]_{p_b\downarrow} \;\middle|\; N_I, \qquad U(I): \quad N_C \;\middle|\; \big[P_b N_b\big]_{p_b\uparrow} \;\middle|\; P_C,$$

the middle term running over the concordant strata in $L(I)$ and over the inverted ones in $U(I)$. *Proof.* The run-exchange argument of Proposition 2 gives the two extremes, an inverted stratum now splitting into its positives, of positive share one, at the bottom and its negatives, of share zero, at the top; and the feasible arrangements, being the linear extensions of one partial order, are joined by adjacent transpositions that move the rank sum by at most one, so every value between is attained. Both steps are below. □

Strict rankings, ties and the rounding band. Theorem 1 is stated for strict rankings at an exact integer S. Tied scores induce midranks, which lie in the permutohedron, the convex hull of the rank vectors, and can give half-integer S; a five-item design with strata $(1, 0, 0)$ and $(1, 0)$ and scores $(1, 2, 3, 4, 4)$ has $S = 5.5$ and pooled AUC $5/12 \approx 0.417$, a value no strict ranking attains to three decimal places. Both objectives are linear, so the dual bound of (4) evaluated at that real S holds for every tied score with that rank sum, and Corollary 1 shows it is attained at every half-integer S by a weak ordering with one tied pair. Two checks confirm this. Exhaustive enumeration of every weak ordering: on the five-item design above (541 orderings, 13 distinct S, six of them half-integer), the six-item validation design, strata $(0, 0, 1, 1, 2, 2)$ and labels $(1, 1, 1, 0, 0, 0)$ (4,683 orderings, 19 values of S), and two seven-item designs (47,293 orderings each, 25 values of S each), the bound

is valid and attained at every S (`ties_enumeration.py`). And the construction of Corollary 1 itself, run at 4,196 half-integer S of 56 random designs of 4 to 397 items (every half-integer S on the 36 designs of at most six items, 200 sampled ones on each larger design), for both bounds: the two relaxed minimisers always differ by one adjacent positive/negative swap, the tied vector has the required rank sum, its objective equals the dual to 4×10^{-11}, and on the 36 designs small enough to enumerate no weak ordering beats it (`ties_corollary_check.py`). The band's real ends are in general not attainable rank sums; the dual there is an outer bound, and on PRM800K it differs from the value at the nearest attainable S by less than 10^{-11} (`real_band_ends.py`), far below printed precision. Over the band the lower bound is convex in S with derivative $\theta^*(S)$ and the upper bound concave, so when θ^* keeps its sign across the band the extreme sits at an end, which is the case on every rule and case we audit (`band_slopes.py`). The identified set lies between the two extremes and contains both where a ranking attains them, though not every value between; the numbers printed in the tables are extremes rounded outward, not attained values. Table 4 tabulates, for each bound, the constructed ranking's pooled AUC and permutation mean against the dual value.

Within-stratum extremes and partly inverted arrangements (Propositions 2–3). `within_extremes.py` enumerates every ranking of 42 random designs of four to seven items and confirms that the rank sums of rankings with $A_{\text{within}} = 1$ are exactly the integers from the block arrangement of Proposition 2 to the maximum, and likewise for $A_{\text{within}} = 0$; it also builds, at the printed 0.617 (PRM800K) and 0.765 (ProcessBench), explicit rankings with within-AUC 1 and 0 inside the band, and the 2,000-stratum design of the text.

Theorem 1, in full. For fixed θ the relaxed minimum pairs the largest adjusted weight $c_i = w_i - \theta y_i$ with the smallest rank. Items with equal c_i form tie classes occupying contiguous rank blocks; a block of m ranks $r_0, \ldots, r_0 + m - 1$ holding k positives gives the positives a rank sum between $\sum_{j<k}(r_0 + j)$ and $\sum_{j<k}(r_0 + m - 1 - j)$, and every integer between is attained by moving one positive up one rank at a time. Summing over blocks, the values of $y^\top R$ over relaxed minimisers form an integer interval $[S^-(\theta), S^+(\theta)]$. D is concave and piecewise linear, its breakpoints lie where a positive and a negative tie in c, and its superdifferential at θ is $[S - S^+(\theta),\ S - S^-(\theta)]$. For θ below all breakpoints the positives hold the smallest ranks and the slope is $S - S_{\min} \geq 0$; above all of them the positives hold the largest and it is $S - S_{\max} \leq 0$; so a maximiser θ^* exists, and 0 lies in the superdifferential there. Hence $S^-(\theta^*) \leq S \leq S^+(\theta^*)$: some relaxed minimiser at θ^* has $y^\top R = S$ exactly. It is feasible and its objective is $D(\theta^*)$. The maximum is the same argument with the sort reversed. □

Why the blocks are contiguous. The exchange argument of Proposition 2 is self-contained and we restate it with the bookkeeping. Cut a feasible list into maximal runs of one stratum's items; adjacent runs belong to different strata, so exchanging two of them preserves each stratum's internal order and hence feasibility. If the lower run has b items with g positives and the upper a items with d positives, the exchange moves the lower run up by a ranks and the upper down by b, changing the positives' rank sum by $ga - db$. At a minimiser no exchange lowers it, so $g/b \geq d/a$ for every adjacent pair: the share of positives is non-increasing run by run. Within one stratum the runs cut the string "negatives then positives" in order, so their shares are non-decreasing. If a stratum is therefore split into consecutive runs u and v, every run w between them has share between them in both directions, so all are equal, the exchanges there change nothing, and u can be walked up to meet v. Each merge lowers the run count, so some minimiser gives every stratum one block, and the adjacent-block exchange of the proposition orders the blocks. `within1_contiguity.py` confirms on all 48 designs of at most eight items that a contiguous minimiser exists, that no minimiser violates the run inequality, and that the named block arrangement attains the minimum.

The same result from scheduling. Read a ranking as a sequence of n jobs, item j in rank position r_j, every job of unit processing time, so that job j completes at time $C_j = r_j$. Give job j weight y_j. Then $\sum_j y_j C_j$ is exactly the positives' rank sum, and the constraint $A_{\text{within}} = 1$ is the precedence relation "every job of N_b before every job of P_b" for each mixed stratum b. That relation is the series composition of two independent (parallel) sets per stratum, and distinct strata are unrelated, so the precedence graph is series–parallel and Lawler (1978)'s algorithm applies, extending Smith (1956)'s ratio rule to precedence constraints: it processes the decomposition tree bottom-up, and at a series node it keeps a preceding set adjacent to its successor whenever the successor has the larger weight-to-time ratio. Here N_b has ratio 0 and P_b ratio 1, so $N_b P_b$ is kept adjacent and behaves as one module of n_b jobs with total weight $n_{+,b}$, that is of ratio p_b; at the top (parallel) node the modules

are sequenced by non-increasing ratio. This gives the contiguous blocks of Proposition 2, and the adjacent-block exchange in its proof re-derives the block order without appealing to the algorithm.

For Proposition 3 the two arrangements are, bottom-up: (min) the positives of I; the concordant strata as blocks $[N_b, P_b]$ with p_b non-increasing; the negatives of I; (max) the negatives of the concordant strata; the inverted strata as blocks $[P_b, N_b]$ with p_b non-decreasing; the positives of the concordant strata. The run-exchange argument above gives both directly. It used only that adjacent runs of distinct strata may be exchanged, which preserves every stratum's internal order and hence feasibility here too, and that at a minimiser the share of positives is non-increasing run by run from the bottom. In a concordant stratum the runs cut "negatives then positives" in order, so their shares are non-decreasing, which forces the runs together into one block of share p_b exactly as in Proposition 2. In an inverted stratum the runs cut "positives then negatives", so their shares are already non-increasing and nothing forces them together; the minimum splits the stratum as far as the order allows, sending P_b, of share 1 and so the largest available, to the bottom and N_b, of share 0, to the top, with the concordant blocks between them in non-increasing p_b. That is $L(I)$, and $U(I)$ is the mirror image. Lawler's rule returns the same two arrangements, an inverted stratum's positives now preceding its negatives so that the earlier part carries the larger ratio and the two sets are sequenced separately, at the ratio-one and ratio-zero ends. That every integer between is attained follows from connectivity rather than from a fixed recipe. The feasible arrangements are exactly the linear extensions of the partial order that puts each concordant stratum's negatives below its positives and each inverted stratum's positives below its negatives. Any two linear extensions of a finite partial order are joined by adjacent transpositions of incomparable elements, each of which is again a linear extension: if L and M first differ at position i and $x = M[i]$ sits later in L, the element just below x in L follows x in M, so it is incomparable to x and the two can be swapped, moving x down one place; repeat. Such a transposition changes the positives' rank sum by $+1$, -1 or 0, by ± 1 when the two elements carry different labels and 0 otherwise. Along a path from a minimiser to a maximiser the rank sum therefore takes every integer between $L(I)$ and $U(I)$, and tying an adjacent pair of different labels gives the half-integers. The path covers swaps between two concordant strata, which a recipe phrased only in terms of the inverted set would miss. `within_family.py` enumerates every ranking of 49 random small designs with a random inverted set I and confirms the interval, and on the four public designs searches nested families of I (prefixes of the mixed strata sorted by p_b or by pair count, each way) by bisection on the prefix length for the largest λ_I whose $[L(I), U(I)]$ meets the printed band; the certified within-AUC is $1 - \lambda_I/\lambda$ for that I.

The unconstrained rank-difference bound, and the number of printed digits. The naive bound of Section 5.1 uses the rank form (3): $\nu = A_{\text{pooled}} - \sum_i (y_i - w_i) R_i/(n_+ n_-)$, and ranging the sum over all rankings (two sorts) while ignoring that the same ranking must also produce A_{pooled}, with A_{pooled} taken at the ends of the same three-decimal rounding band, then intersecting with the annotation-only interval, yields $[0.560, 0.910]$ for the ORM rule and $[0.236, 0.910]$ for the first-step rule (rounded outward, as in Table 1; the unrounded values are $[0.5609, 0.9092]$ and $[0.2369, 0.9092]$), against the sharp bounds $[0.562, 0.908]$ and $[0.294, 0.838]$. At the top of the scale the naive bound is nearly as tight; lower down, the sharp bounds remove about 0.06 at the lower end and 0.07 at the upper. Table 3 reports both bounds on a fixed grid of printed values, 0.50 to 0.99 in steps of 0.01, on the four designs whose outcomes and stratum keys are public (`naive_vs_sharp_sweep.py`); every grid value is in the released file, and on every band θ^* keeps its sign, so the extremes sit at the band's ends. The pattern is empirical and design-specific. On PRM800K, ProcessBench and WikiBio the reduction in width was largest near a printed 0.5 in the range examined (0.15–0.21), 0.04–0.05 at 0.75 and under 0.01 above 0.85, and the verdict on 0.5 changed only on PRM800K, at the grid value between the two thresholds; by the symmetry of Section 4 the same holds below 0.5. On MMLU the largest reduction, 0.09, occurred at 0.82, and no printed value on the grid excluded 0.5 under either bound. The reduction depends not on λ alone but on the restriction that the stratum sizes, their positive rates and the reported figure jointly place on the ranking, so the sweep documents each design's behaviour and does not establish a rule that predicts the reduction from λ. The number of printed digits matters little: the ORM bounds at two, three and four decimal places are $[0.556, 0.909]$, $[0.562, 0.908]$ and $[0.562, 0.908]$, and the first-step bounds $[0.293, 0.841]$, $[0.294, 0.838]$ and $[0.295, 0.838]$ (rounded outward) (`rounding_and_naive.py`).

The procedure of Section 4 in detail. (i) A ternary search for θ^* on $[-2, 2]$, 80 iterations, D being concave; (ii) a snap to the breakpoint $p_a - p_b$ with the best dual value among the eight nearest the search value; (iii) a stable sort of $c(\theta^*)$, tie classes at tolerance 10^{-12}, and a block-by-block

Table 3: Unconstrained rank-difference (naive) bound against the sharp bounds of Theorem 1 on a grid of printed pooled AUCs, four public designs. "thr." is the smallest grid value at which the bound excludes $\nu = 0.5$ (none: no grid value up to 0.99 does); the annotation-only intervals are those of Appendix C; "width cut" is the naive width minus the sharp width at the printed value shown; "max" its largest value on the grid and where it occurs. Bounds are over the three-decimal rounding band.

design	λ	thr. naive/sharp	width cut at 0.55 / 0.75 / 0.95			max (at)
PRM800K	0.00105	0.89 / 0.88	0.185	0.047	0.003	0.204 (0.52)
ProcessBench MATH	0.00054	0.84 / 0.84	0.147	0.048	0.000	0.153 (0.59)
WikiBio	0.00236	0.90 / 0.90	0.173	0.038	0.003	0.208 (0.51)
MMLU	0.01507	none / none	0.027	0.065	0.039	0.090 (0.82)

assignment of the positives' ranks that starts at each block's minimum and raises the sum by the $S - S^-(\theta^*)$ still needed; (iv) a check that the result is a permutation with $y^\top R = S$, whose objective is reported beside $D(\theta^*)$. The construction matches exhaustive enumeration at all 230 attainable values of nineteen designs of up to nine items, at 4,196 half-integer rank sums of 56 random designs of 4 to 397 items and every weak ordering of 40 small designs, and agrees with the dual to 5×10^{-13} on synthetic designs up to $n = 100{,}000$.

The numerical certificate. Table 4 lists, for each bound of Table 1, the rank sum S at which the extreme is attained (an end of the printed value's rounding band), the multiplier θ^* found by ternary search and snapped to the nearest breakpoint $p_a - p_b$ (eight candidates, ties judged at 10^{-12}), the pooled AUC and permutation mean of the constructed ranking, which is asserted to be a permutation with $y^\top R = S$ exactly, and the dual value at that θ^*. The difference is floating-point noise.

Table 4: Primal–dual certificate: constructed ranking against dual bound, PRM800K, at the printed values of Table 1. S is the integer rank sum nearest the band's end (the real end, e.g. $249{,}592{,}268{,}160.13$ for ORM at 0.9405, is not an attainable rank sum); each row is a valid bound (the dual at that S) with an attaining strict ranking. "pooled" is the constructed ranking's own pooled AUC; "attained" its permutation mean; "dual" the value of (4) at θ^*. The dual at the real ends differs from these values by less than 10^{-11} (`real_band_ends.py`).

rule	side	S	pooled	θ^*	attained	dual	difference
ORM	lo	249,592,268,161	0.9405	+0.9574	0.562150	0.562150	-1×10^{-15}
ORM	hi	249,592,268,161	0.9405	-0.1001	0.907733	0.907733	$+5 \times 10^{-15}$
released	lo	248,101,181,669	0.9315	+0.9505	0.553565	0.553565	-8×10^{-16}
released	hi	248,101,181,669	0.9315	-0.0664	0.908479	0.908479	$+3 \times 10^{-15}$
prod	lo	246,444,418,900	0.9215	+0.9428	0.544100	0.544100	-3×10^{-15}
prod	hi	246,444,418,900	0.9215	-0.0335	0.908975	0.908975	$+2 \times 10^{-15}$
min	lo	246,278,742,624	0.9205	+0.9420	0.543157	0.543157	$+3 \times 10^{-15}$
min	hi	246,278,742,624	0.9205	-0.0304	0.909007	0.909007	$+3 \times 10^{-15}$
last	lo	245,947,390,070	0.9185	+0.9402	0.541275	0.541275	-4×10^{-15}
last	hi	245,947,390,070	0.9185	-0.0244	0.909062	0.909062	$+1 \times 10^{-15}$
mean	lo	242,302,511,979	0.8965	+0.9227	0.520784	0.520784	-1×10^{-15}
mean	hi	242,468,188,255	0.8975	+0.0263	0.909017	0.909017	$+8 \times 10^{-15}$
first	lo	195,913,154,460	0.6165	+0.6844	0.294689	0.294689	-1×10^{-15}
first	hi	196,078,830,735	0.6175	+0.4419	0.837376	0.837376	0

Ties and the certificates. The bound of Section 4 is stated over permutations of $1, \ldots, n$, while a score with ties induces midranks. A midrank vector is the average of the rank vectors of the strict orderings compatible with the tie pattern, so it lies in the permutohedron, the convex hull of those permutations. Both A_{pooled} and $\mathbb{E}[A_{\text{pooled}} \mid H_0]$ are linear in the rank vector, and a linear function over a polytope attains its minimum at a vertex, so relaxing the feasible set from permutations to the permutohedron leaves the dual bound unchanged: the bound covers tied scores, and by Corollary 1 it is attained at half-integer rank sums by a weak ordering with one tied pair. Two certificates follow. Table 4 certifies the strict-score bounds at the integer rank sum nearest each band end. Once ties are admitted the attainable rank sums are $\frac{1}{2}\mathbb{Z}$ and the nearest attainable rank sum inside the band can be a half-integer (for the ORM lower end the band's real end is $S = 249{,}592{,}268{,}160.13$, the nearest

half-integer inside it 160.5, the nearest integer 161); Table 5 certifies the bound over weak orderings at that rank sum with the tied construction, and reports its difference from the strict certificate, which is below 10^{-11} throughout and leaves every printed bound unchanged (`certificate_tied.py`). Ties are in any case under 0.005% of cross-label pairs on every PRM800K rule. The bounds are the interval hull of the identified set, not the set: on the five-item design above, 48 of the 541 weak orderings have pooled AUC exactly $5/12$, their ν runs from $5/12$ to $7/12$, and none has $\nu = 0.5$ (the nearest values are 0.486 and 0.514; `membership_counterexample.json`). The bounds alone therefore do not determine membership: a value inside them is attained only when a witness shows it.

Table 5: Certificate over weak orderings, PRM800K, at the attainable rank sum $S \in \frac{1}{2}\mathbb{Z}$ nearest each band end (a half-integer where the band's real end lies between an integer and a half-integer; the strict certificate of Table 4 uses the nearest integer). "kind" is the witness: a strict ranking (Theorem 1) or a weak ordering with one tied adjacent pair (Corollary 1). "gap" is attained minus dual; "vs strict" is the attained value minus the strict certificate's, all below 10^{-11}, so every printed bound is unchanged.

rule	side	S	kind	attained	dual	gap	vs strict
orm	lo	249,592,268,160.5	tied	0.562150172	0.562150172	-4×10^{-15}	-3×10^{-12}
orm	hi	249,592,268,160.5	tied	0.907733498	0.907733498	-4×10^{-15}	$+3 \times 10^{-13}$
released	lo	248,101,181,668.5	tied	0.553565062	0.553565062	$+3 \times 10^{-15}$	-3×10^{-12}
released	hi	248,101,181,668.5	tied	0.908478678	0.908478678	$+7 \times 10^{-16}$	$+2 \times 10^{-13}$
prod	lo	246,444,418,900	strict	0.544099509	0.544099509	$+2 \times 10^{-15}$	0
prod	hi	246,444,418,900	strict	0.908975447	0.908975447	$+1 \times 10^{-15}$	0
min	lo	246,278,742,623.5	tied	0.543157119	0.543157119	$+8 \times 10^{-15}$	-3×10^{-12}
min	hi	246,278,742,623.5	tied	0.909007360	0.909007360	$+2 \times 10^{-16}$	$+9 \times 10^{-14}$
last	lo	245,947,390,069.5	tied	0.541274958	0.541274958	-3×10^{-15}	-3×10^{-12}
last	hi	245,947,390,069.5	tied	0.909062104	0.909062104	-2×10^{-15}	$+7 \times 10^{-14}$
mean	lo	242,302,511,979	strict	0.520784011	0.520784011	-6×10^{-16}	0
mean	hi	242,468,188,255	strict	0.909017152	0.909017152	0	0
first	lo	195,913,154,459.5	tied	0.294689209	0.294689209	$+4 \times 10^{-15}$	-2×10^{-12}
first	hi	196,078,830,735.5	tied	0.837375983	0.837375983	-4×10^{-16}	$+1 \times 10^{-12}$

B Relation to known results and additional related work

Stratified and clustered estimators. The grouped AUC of Section 2 weights users by impressions rather than by within-user pair count (Appendix G), and either way it is computed where the scores are. van Elteren (1960) combines Wilcoxon statistics across strata with fixed size-based weights, the rank-sum expectation behind (2); Obuchowski (1997) develops estimation and inference for clustered ROC curves that account for within-cluster correlation, without restricting to within-cluster pairs. Appendix G compares (5) with the unweighted mean of per-stratum AUCs, which answers a question about a typical stratum rather than a typical pair. Survival analysis restricts the concordance index to within-stratum pairs for exactly the reason (1) gives (Harrell et al., 1996; van Houwelingen & Putter, 2011). The general hazard is aggregation bias (Simpson, 1951). The classical AUC literature is about other properties: Bamber (1975) relates the area to the Mann–Whitney statistic and gives its variance, DeLong et al. (1988) compare correlated AUCs, and Hand (2009) criticises the AUC's coherence as a summary; none concerns its reference point.

Relation of Theorem 1 to known results. The proof of Theorem 1 uses only that y is binary and that ranks are consecutive integers; w may be any real vector, so the statement is a general one about sorting under a single rank-sum constraint with 0/1 coefficients. In polyhedral terms: the edges of the permutohedron join permutations that differ by swapping two adjacent ranks (Ziegler, 1995), along which $y^\top x$ changes by at most one, so at integer S the hyperplane $y^\top x = S$ cuts no edge at an interior point: it either misses an edge, meets it at a vertex, or contains it, and in each case the section of the permutohedron has only permutations as vertices and is an integral polytope; the linear programme over the section therefore has a permutation as an optimum, which is Theorem 1, and Corollary 1 is the same observation at half-integer S, where the section's vertices are edge midpoints. For a general assignment problem with one side constraint the Lagrangian dual has a gap (Geoffrion, 1974); here the product structure of both the objective and the constraint removes it. We searched the literature on rearrangement inequalities (Hardy et al., 1934), on side-constrained assignment and on

the permutohedron and found no prior statement of this integrality, but the argument is elementary and we do not rest the contribution on its priority: what we claim is its use for the identification problem and the constructive algorithm of Section 4. Proposition 2 is, by contrast, the unit-time, 0/1-weight instance of a known result: minimising a weighted sum of positions under series–parallel precedence constraints is solved by Lawler (1978)'s module ordering, of which the block arrangement is that instance. We prove it directly rather than by citation, the special structure admitting a short exchange argument (Section 4 and Appendix A).

Concurrent audits. Bulut (2026) reports, on LLM reasoning trajectories, a trace-blind difficulty proxy at pooled AUROC 0.873, and in a separate reconstruction of a published early-window result a pooled 0.849 whose within-problem counterpart is 0.496 under that paper's pre-registered failure-count weighting and 0.515 under the pair weighting used here, neither distinguishable from chance; Chen et al. (2026) note the same confound for reference-free judge gates, and a second audit of weakly supervised video anomaly detection (Abdulaziz & Bondarev, 2026) reads fixed frame scores under three pairing granularities and finds that pooled AUC does not predict within-video localisation, replicating the effect Song & Lee (2026) report. HalluScan (Cherif, 2026) reports a detection AUROC pooled over 600 responses from four models and three domains beside per-configuration AUROCs, and notes that the pooled ranking of detectors differs from the macro-averaged one; with the stratified figures reported, the identification question of Section 4 does not arise there, and we did not adopt it as a second case because its labels and code were unreleased at the time of writing. Both of those audits re-read scores they hold; our question begins where the scores are unavailable (below).

Adjacent lines. Tschalzev et al. (2026) make a different complaint about aggregation, that averaging across datasets hides which models are irreplaceable on any one of them; their target is model selection, ours what a single figure identifies. A separate literature asks whether the label a score is validated against stands in for the construct at all, studied for routing signals and failure-attribution benchmarks; content-blind probes of manipulation benchmarks (Jiang et al., 2026) and hypothesis-only baselines in NLI (Gururangan et al., 2018; Poliak et al., 2018) report accuracy, which Appendix C places outside our scope. A distributional critique of binary success rates on real-robot benchmarks (Arkhangelskiy, 2026) and a result that deployment gain follows a conditional-regret functional rather than AUC (Sun & Yang, 2026) concern what a ranking statistic is worth downstream. All of these ask what the comparison is made *against*, or what it buys; ours asks what the unreleased scores' permutation mean could be.

The artifacts. Lightman et al. (2024) released the scored samples of Section 5, Wang et al. (2024) Math-Shepherd, and Manakul et al. (2023) the WikiBio annotations. Vashurin et al. (2025) benchmark 35 uncertainty methods and release the per-instance estimates Section 5 re-analyses; their headline metric is a rejection-curve summary rather than a pairwise statistic, so Section 5 measures the exposure on their released scores and does not re-read the number they report.

C THE WEIGHT λ AND THREE SCOPE CONSTRUCTIONS

What is out of scope. The exposure is specific to threshold-free pairwise statistics. The mechanism studied here is the cross-stratum pair: it exists only for statistics defined over pairs of examples, so pooled accuracy, F1 and calibration error are outside it, whatever aggregation issues they have of their own. Where the stratum determines the outcome, as in out-of-distribution detection with the corpus as stratum, every stratum is pure, $\lambda = 0$, and the pooled figure is the cross-stratum term that design intends.

Remark (the weight). For B strata of equal size and equal outcome rate, $\lambda = 1/B$.

Proof. The within-stratum pair count is $B(np/B)(n(1-p)/B) = n^2p(1-p)/B$ against a total of $n^2p(1-p)$. □

Both halves of the hypothesis matter. Equal size alone gives $\lambda = (1/B)\big(1 - \mathrm{Var}(p_b)/[\bar{p}(1-\bar{p})]\big)$, since $\mathbb{E}[p_b(1-p_b)] = \bar{p}(1-\bar{p}) - \mathrm{Var}(p_b)$: dispersion in the stratum outcome rates depresses λ below $1/B$ by exactly that factor. Unequal strata can push λ above $1/B$ — three strata with $(n_{+,b}, n_{-,b})$ of $(100, 100)$, $(1, 1)$ and $(1, 1)$ give 0.96 — or below it, as when strata whose examples are all correct or all wrong contribute no within-stratum pairs (Section 5). Section 5's equal-sized design checks the identity: 57 subjects of exactly 100 questions but far from equal-rate give $\lambda = 0.01507$ where

$1/B = 0.01754$, which the identity reproduces to five decimals. Table 6 gives the annotation-only endpoints each public design's counts imply.

Table 6: What the released annotations alone determine, through the per-stratum counts $(n_{+,b}, n_{-,b})$ they imply. PRM800K and the two LM-Polygraph runs release per-example scores; Math-Shepherd releases the step labels its prefix scores are built from, the WikiBio release carries annotations only, and ProcessBench carries the first-error annotations its per-step outcomes follow from (Appendix E). Endpoints are the sorted-block construction of Proposition 1.

benchmark	unit in stratum	n	strata	λ	null-reference interval
PRM800K	sample in problem	815,632	500	0.00105	[0.090, 0.910]
Math-Shepherd	solution to a problem	7,039	400	0.00172	[0.103, 0.897]
LM-Polygraph CoQA	question about a story	2,000	486	0.00212	[0.217, 0.783]
SelfCheckGPT WikiBio	sentence in a passage	1,908	238	0.00236	[0.096, 0.904]
LM-Polygraph MMLU	question in a subject	5,700	57	0.01507	[0.292, 0.708]
ProcessBench MATH	step in a solution	6,505	1,000	0.00054	[0.066, 0.934]

C1. Four strata with per-stratum base rates from 0.1 to 0.9 and per-stratum accuracies from 0.30 to 0.95. Pooled accuracy and $\sum_b (n_b/n)\mathrm{Acc}_b$ agree to $0.00\mathrm{e}{+00}$; the same data's AUC has $\lambda = 0.135$, pooled 0.648, within 0.690, cross 0.642.

C2. 500 in-distribution and 500 out-of-distribution examples with stratum equal to corpus, so each stratum holds one outcome only and no within-stratum pair exists. $\lambda = 0$ exactly; there is no within term to report and the pooled value 0.802 is the intended estimand.

C3. Twenty strata of 100 examples with outcome rates swept from 0.05 to 0.95 and scores drawn independently of the outcome inside every stratum, varying only the alignment between stratum score means and stratum outcome rates. Pooled ranges 0.184–0.821 across the sweep while within stays in 0.480–0.506.

D PRM800K TABLES, MATH-SHEPHERD AND PROTOCOL DETAILS

Table 7 gives the seven rules of Table 1 with their best-of-N accuracies; Table 8 gives the pooled AUC, the permutation mean ν and A_{within} for every prefix span. A prefix score over the first k steps is the product of the released per-step correctness probabilities over those steps, k being $\lceil fL \rceil$ for span f and solution length L; "step 1 only" is the first step's probability; the released `prm_score` and the ORM comparator are OpenAI's solution-level scores as released.

The first-step rule. Its within-problem AUC of 0.526 is small but has a cluster-bootstrap interval of $[0.516, 0.536]$ over the 481 problems holding both outcomes, which excludes 0.5 (`first_step_within_ci.py`). Selecting each problem's top-scored sample from its full candidate pool, 1,631 samples on average, beats a random pick by 0.024 with interval $[-0.007, 0.056]$, so this rule's selection gain is not distinguishable from zero (`bon_gain_ci.py`).

λ = 0.00105, about 1/952. It falls below 1/500 because problems whose samples are all correct or all wrong contribute no within-stratum pairs, and unequal problem sizes move it as well. By identity (1)

Table 7: Seven scoring rules from OpenAI's PRM800K release, ordered by the pooled AUC. Best-of-N accuracy is averaged over problems, with 200 resamples of the N-subset per problem for finite N.

rule	pooled	within	BoN 4	BoN 16	BoN 64	BoN 256	BoN all
ORM comparator	0.941	0.884	0.631	0.694	0.717	0.722	0.724
released `prm_score`	0.932	0.881	0.633	0.707	0.746	0.768	**0.782**
product over steps	0.922	0.871	0.629	0.705	0.739	0.764	0.774
minimum step	0.921	0.868	0.629	0.702	0.739	0.760	0.778
last step	0.919	0.867	0.623	0.689	0.714	0.726	0.736
mean step	0.897	0.844	0.616	0.694	0.733	0.755	0.772
first step	0.617	0.526	0.490	0.498	0.496	0.502	0.506

Table 8: PRM800K prefix scores against the within-problem null. $\lambda = 0.00105$ throughout, so the cross-problem term is within 0.0001 of the pooled figure on every row and its column is omitted. The null is equation (2), evaluated in closed form; the mean of 2,000 simulated within-problem label permutations is within 0.001 of it at every row.

score	pooled	permutation mean	A_{within}
step 1 only	0.617	0.607	0.526
prefix 25%	0.738	0.693	0.612
prefix 50%	0.819	0.738	0.703
prefix 75%	0.877	0.760	0.798
product over steps	0.922	0.783	0.871
released `prm_score`	**0.932**	**0.791**	**0.881**
ORM comparator	0.941	0.797	0.884

Table 9: Math-Shepherd prefix scores, 400 GSM8K problems: the score over span f is the mean of the released step labels over the first $\lceil fL \rceil$ steps. The full-span row is degenerate (see the Math-Shepherd paragraph of this appendix) and excluded from every claim.

prefix span	λ	pooled	within	null mean	null 95%
steps 1..25%	0.00172	0.726	0.651	0.636	[0.629, 0.644]
steps 1..50%	0.00172	0.850	0.779	0.694	[0.685, 0.702]
steps 1..75%	0.00172	0.958	0.924	0.735	[0.725, 0.744]
steps 1..100%	0.00172	1.000	1.000	0.749	[0.739, 0.758]

the cross-problem term exceeds the pooled figure by $\lambda(A_{\text{cross}} - A_{\text{within}})$, which is between 0.00005 and 0.0001 on the seven rules; the two can still round differently at the third decimal, as for the first-step rule (pooled 0.617, cross 0.618).

For the released `prm_score`, the within-problem permutation null has expected pooled AUC 0.791. The observed 0.932 lies outside the permutation interval. That is a statement about where the observed value falls relative to the distribution; reading it as evidence about within-problem information needs the exchangeability condition stated below, and either way the comparison does not quantify within-problem discrimination, since the permutation moves the cross term as well (Section 3). The amount of discrimination is $A_{\text{within}} = 0.881$, reported beside it. What the permutation mean settles is the scale on which 0.932 should be read, and the artifact does not state which scale a reader is meant to use.

On Math-Shepherd (Table 9), $\lambda = 0.00172$, about 1/581, and the reference rises from 0.636 to 0.735 across the usable spans. The full-span row is degenerate: its last step's label is the outcome itself, so the score contains what it would predict and separates the outcomes perfectly in the released data. It is excluded from every claim.

The mechanism that puts λ below $1/B$ on PRM800K reproduces here on a second artifact with a different label source: 125 of the 400 problems are solved by every sampled solution or by none, contribute no within-stratum pair, and so push 0.00172 below $1/400$. The null-reference envelope of Proposition 1 is $[0.103, 0.897]$ (printed rounded outward) over the same 400 strata (1,784 successes, 5,255 failures), computed by `scripts/mathshepherd_envelope.py` with the loader that produces λ, so the parse is the one reported above. Sorted blocks are extremal against 2,000 random orderings, and the interval contains 0.5.

Alternative within-stratum estimator. The pair-weighted within term (5) answers a question about a typical same-stratum pair. The unweighted mean of per-stratum AUCs, the statistic a careful reader might substitute, answers a question about a typical stratum; its within-stratum permutation null is also 0.5. We prefer the pair-weighted form because it is the term appearing in (1).

The two null expressions check against each other. Under H_0 the within-stratum term has expectation 0.5, so (1) gives $\mathbb{E}[A_{\text{pooled}} \mid H_0] = \lambda/2 + (1-\lambda)\,\mathbb{E}[A_{\text{cross}} \mid H_0]$; equating with (2) determines the cross-stratum term. On the released PRM800K score, (2) gives 0.79052; substituting

that into (1) implies a cross-stratum null of 0.79082, and the two agree, since the identity requires the cross-stratum term to exceed the pooled one by exactly $\lambda(\mathbb{E} - \frac{1}{2})/(1-\lambda) = 0.00030$.

Conditioning on the fit. Holding the fitted scores fixed conditions on the fit rather than propagating fitting variance. Two uses of the permutation must then be kept apart. Equation (2) evaluated on fixed scores is a descriptive reference: the pooled AUC those scores would attain on average if outcomes were exchangeable within strata, with the scores and the stratum memberships held fixed while the pair labels move. Treating the same permutation distribution as a valid test is a further step, and out-of-fold scoring does not by itself license it: an example's outcome enters the models fit on the other folds, so exchangeability of outcomes given the scores is an assumption and not a consequence of the fold structure. We use the reference throughout, and no conclusion turns on a value narrowly clearing a permutation interval.

E CASE STUDY: PROCESSBENCH MATH

The scored artifacts of Section 5 release per-example scores, which is what lets us measure their permutation means; the WikiBio release carries annotations only. This case is the configuration the identification question is about: a published pooled figure whose scores are not released, over a benchmark whose outcomes and stratum keys are public.

Table 10: What a reader of Lee et al. (2026) can say about the reference point of its published figures — every row of its Table 3. "Sharp bounds" are the bounds of Theorem 1 over the three-decimal rounding band, read as a standard AUC: valid bounds attained by explicit strict rankings at the feasible rank sums nearest the band's ends, so a value outside one is attained by no score reproducing the figure. The annotation-only interval is $[0.066, 0.934]$ for every row (`processbench_case_study.py`, `exact_region.py`). Rand_Merge is that paper's control, a model trained on randomised labels rather than an uninformative score, so its 0.379 is not a measured permutation mean.

labelling method	reported ROC-AUC	sharp bounds	bounds exclude 0.5?
MC	0.759	[0.429, 0.899]	no
PAV	0.757	[0.427, 0.899]	no
CPMI	0.765	[0.434, 0.901]	no
CPMI_Merge	0.766	[0.435, 0.902]	no
Rand_Merge	0.379	[0.162, 0.682]	no

The paper is Lee et al. (2026), which trains process reward models under a new step-labelling rule and reports a step-level ROC-AUC on ProcessBench's MATH split. Two facts make the design recoverable, both from its released evaluation code (`inference/pb_plot_main.py`): the statistic is one AUC over the steps of every instance concatenated, so the strata are the solutions; and the per-step outcome follows from ProcessBench's gold first-error index, every step before it positive and the rest negative, which its process-level evaluation section states in prose as well. That section takes the higher ROC-AUC of one labelling method over another as indicating a stronger ability to "rank correct steps above incorrect ones", and describes the statistic as a threshold-free measure of step-level discrimination. We do not read this as a claim that the estimand is within-solution: the surrounding text also describes separation in the global score distribution, and the phrase "step-level" does not by itself fix which pairs are meant. Our question is the reader's. A reader comparing steps within one solution wants the within-solution reading, and the ROC-AUC of its Table 3 is the only threshold-free number offered for it; its Table 4 reports ProcessBench F1 at a chosen threshold, which does not identify a within-solution AUC either. Our analysis is confined to what the Table 3 figure supports. What follows asks how much of that reading the reported AUC and the annotations support, and the answer does not depend on which estimand the authors intended. Reconstructing the design from the benchmark gives 6,505 steps in 1,000 solutions, 3,772 positive and 2,733 negative. Of the solutions 521 are pure — 406 with no error at all and 115 whose first error is the first step — and contribute no within-solution pair, so $\lambda = 0.00054$, the smallest in this paper. Sorted blocks are extremal against 2,000 random orderings and the annotation-only interval is $[0.0669, 0.9331]$, printed $[0.066, 0.934]$.

The regions of Table 10 are wide, they contain 0.5, and their upper ends lie above the reported figures. A null reference above an observed pooled AUC does not by itself put the within-stratum AUC below 0.5, because the permutation moves the cross-stratum term as well (Section 3), so we construct the ordering and measure it instead. A score reproducing 0.765 exists whose within-solution AUC is 0.0000 over the 5,569 same-solution pairs: it inverts the correct/incorrect order in all 479 solutions that contain both, recovers the published figure entirely from the ordering across solutions, and has a permutation mean of 0.8997.

The other side of the scale is reachable too, and by a position-only score that uses no step text. Rank the steps of every solution by position, earlier above later — the labels are a prefix of correct steps followed by incorrect ones, so this separates all 479 solutions that contain both, for a within-solution AUC of 1.0000 and a pooled AUC of 0.7150. Adding a per-solution constant offset — an ordering of the solutions built from their positive rates, that is from the labels, mixed with a random ordering by a single scalar so that the cross-solution term carries the remaining credit — matches the published 0.765 to its three decimals while leaving the within-solution AUC at 1.0000. (An empirical AUC over $3{,}772 \times 2{,}733$ pairs cannot equal 0.765 exactly; "matches" here and below means agreement at the published precision, the rounding band of Section 4.) So 0.765 is consistent with within-solution AUC 0 and with within-solution AUC 1. The offsets use the labels, so the second witness is an existence construction for non-identification and not a content-blind verifier a reader could deploy; the plain position score, which uses only step position, already reaches within-solution AUC 1 at pooled 0.715. The published number therefore rules out neither extreme of the discrimination it is read as reporting. None of this says the scored model discriminates badly, or well; it says the pooled AUC and the per-stratum outcome counts alone do not decide between the two. Proposition 2 says in advance that both ends are attainable, since 0.765 lies inside $[0.067, 0.933]$; the two constructions exhibit orderings of these 6,505 steps that attain them.

The same holds for the comparison the four labelling variants are reported to support. Their published figures — CPMI_Merge 0.766, CPMI 0.765, MC 0.759, PAV 0.757 — span 0.009, and we build two families of step scores that match all four at their published precision (largest deviation below 0.00005). In each family the within-solution order of a variant is set by inverting whole solutions until its within-solution AUC reaches a chosen value, and the cross-solution order is then interpolated between the p_b-sorted ordering and a random one until the pooled AUC meets the published target. Family A gives within-solution AUCs of 0.950, 0.900, 0.850, 0.800 to CPMI_Merge, CPMI, MC and PAV — the pooled ordering — and family B gives 0.800, 0.850, 0.900, 0.950, exactly the reverse. The constructions show that at these figures both the reported ordering of the variants and its reverse reproduce the four AUCs at printed precision; they do not show that every combination of within-solution AUCs is. The conditions are those of Propositions 2–3, not the size of λ alone. A reader comparing the four on the published numbers is comparing figures that, by pair weight, are almost entirely cross-solution orderings, and the reported AUCs and the annotations do not say which variant discriminates better inside a solution. `processbench_ranking_reversal.py` builds both families.

Why within-solution AUC 1 does not locate the first error. Under the prefix labelling a solution's positives are exactly the steps before its gold first-error index. A score that decreases with step position therefore ranks every positive above every negative of that solution, whatever the index is: on a solution whose correct prefix is two steps long and on one whose prefix is seven steps long the same score attains within-solution AUC 1, while the two first-error indices it would have to report differ. Within-solution AUC fixes which pairs are compared; it does not ask where the boundary between them falls, which is what ProcessBench's F1 scores.

On the position-only point of Section 5.2, the content-blind baselines a within-solution claim would need beside it are step index, relative position and step length. Nothing here says the paper's authors computed anything incorrectly; the case is legible precisely because their evaluation code is public. One condition attaches. The released ROC routine (`_roc_curve_np` and `_auc_trapz` in `inference/pb_plot_main.py`, commit `c03b414f`) departs from the Mann–Whitney value that constraint (3) uses in two ways: it places each tied-score block's ROC point at the block's first element rather than at its end, and it re-sorts the false-positive-rate vector with an unstable sort before the trapezoid integration, which can reorder ROC points sharing a false-positive rate even when no scores tie. On a 15-item tie-free example the two routines give 0.5556 and 0.5972 (`processbench_evaluator_check.py`). We could not check which revision or environment produced the figures of Table 3, so the analysis conditions on the reported figure read as a standard

empirical AUC. The three witnesses of Table 2 were evaluated under both routines and give the same value under each, 0.7650, 0.7150 and 0.7650, so the constructions match the published figure under either reading; the bounds on ν are stated for the standard reading only. What the case shows is what the released information leaves a reader able to conclude, which is a limit of what the released summaries determine. Releasing the three numbers of equation (2) per solution would identify the permutation mean; the per-step scores, or per-solution AUCs with pair counts, would also identify the within-solution AUC.

F THE FULL LM-POLYGRAPH MMLU TABLE

Method provenance. The estimator names are LM-Polygraph's. The methods Section 5 discusses come from `SAR` (Duan et al., 2024), the sequence-level relative Mahalanobis distance (Ren et al., 2023), and `PPLRMDSeq`, which is the hybrid ranking aggregation of Vazhentsev et al. (2023) instantiated for generation by LM-Polygraph; the toolkit attributes no origin paper to `MaximumSequenceProbability`, and we treat it as its baseline.

The artifact. Section 5 and the CoQA check read LM-Polygraph's released managers for their TACL benchmark's StableLM 12B runs (`polygraph_tacl_stablelm12b` and its CoQA counterpart), so the permutation means are properties of that model's error pattern across subjects.

Positive class. For the MMLU analysis in Section 5 an incorrect answer is the positive class, following this literature. Flipping only the outcome labels while holding the score orientation fixed maps each AUC and each ν to one minus itself and leaves the sign-discordant pairs identical.

Sign-discordant pairs. Over all 595 method pairs, 47 have a pooled difference of one sign and a within-subject difference of the other. Selected on that discordance and uncorrected for 595 comparisons, the subset supports no inference and we draw none; one instance is `PPLRMDSeq_decoder` minus `SAR`, twenty-eighth against tenth under the pooled figure, whose pooled difference is -0.092 against a within-subject $+0.023$. A pre-specified explanation for which methods should rise failed on CoQA (Reddy et al., 2019) and we report the failure.

Table 11 lists all 35 uncertainty methods whose released estimates are finite over the 5,700 MMLU questions, ordered by the pooled statistic. $\lambda = 0.0151$ throughout; the remark of Appendix C gives $1/57 = 0.0175$ for a balanced design, and the all-error subject contributes to that gap — it supplies no within-subject pairs — but accounts for only 15% of it; the rest is dispersion in error rates across the other subjects (Section 3).

Table 11: The 35 methods of LM-Polygraph's released MMLU run. Kendall's τ between the two rankings is 0.842; 26 of 35 methods change rank.

method	pooled	null reference	A_{within}	rank pooled	rank within
MaximumSequenceProbability	0.793	0.624	0.733	1	1
CCP	0.752	0.595	0.711	2	2
Perplexity	0.743	0.607	0.681	3	3
TokenSAR	0.743	0.607	0.681	4	4
PPLMDSeq_decoder	0.742	0.607	0.681	5	5
SentenceSAR	0.738	0.605	0.676	6	6
LexicalSimilarity_BLEU	0.711	0.594	0.651	7	13
DegMat_NLI_score_entail	0.708	0.582	0.659	8	10
LexicalSimilarity_rougeL	0.708	0.587	0.654	9	11
SAR	0.707	0.592	0.650	10	14
LexicalSimilarity_rouge1	0.706	0.586	0.653	11	12
FisherRao	0.703	0.567	0.670	12	8
DegMat_Jaccard_score	0.702	0.586	0.649	13	15
RenyiNeg	0.702	0.567	0.668	14	9
EigValLaplacian_NLI_score_entail	0.686	0.571	0.643	15	17
DegMat_NLI_score_contra	0.683	0.566	0.644	16	16
Eccentricity_NLI_score_contra	0.677	0.575	0.626	17	20
MeanPointwiseMutualInformation	0.671	0.578	0.628	18	19
Eccentricity_NLI_score_entail	0.669	0.580	0.612	19	21
Eccentricity_Jaccard_score	0.669	0.581	0.611	20	22
EigValLaplacian_NLI_score_contra	0.664	0.557	0.630	21	18
EigValLaplacian_Jaccard_score	0.661	0.574	0.609	22	23
NumSemSets	0.650	0.563	0.608	23	24
SemanticEntropy	0.648	0.575	0.594	24	26
MeanConditionalPointwiseMutualInformation	0.622	0.538	0.603	25	25
MonteCarloNormalizedSequenceEntropy	0.620	0.569	0.566	26	27
MonteCarloSequenceEntropy	0.618	0.567	0.566	27	28
PPLRMDSeq_decoder	0.616	0.509	0.673	28	7
MeanTokenEntropy	0.553	0.548	0.528	29	29
RDESeq_decoder	0.552	0.552	0.497	30	31
MahalanobisDistanceSeq_decoder	0.467	0.470	0.486	31	32
LexicalSimilarity_rouge2	0.429	0.469	0.452	32	33
RelativeMahalanobisDistanceSeq_decoder	0.413	0.412	0.501	33	30
PTrueSampling	0.305	0.417	0.367	34	35
PTrue	0.287	0.387	0.381	35	34

G CHOICE OF WITHIN-STRATUM ESTIMATOR

Two estimators answer different questions about within-stratum discrimination. The pair-weighted average (5) is the within-stratum term of identity (1) itself: it asks about a typical same-stratum *pair*. The unweighted mean of per-stratum AUCs asks about a typical *stratum*. Both have within-stratum permutation reference 0.5, and both use only strata with two outcomes.

The two estimands differ only when the per-stratum AUCs are heterogeneous, and then they can differ a lot. With eight strata sharing a true within-stratum AUC of 0.750 the two agree in expectation and differ only in spread: over 2,000 draws the pair-weighted estimator has bias -0.0001 and standard deviation 0.028, the unweighted mean -0.0005 and 0.045. Give the two largest strata a true AUC of 0.90 and the rest 0.60, however, and they estimate different quantities: the pair-weighted estimand is 0.881 and the unweighted 0.675, the estimators recover 0.882 and 0.674, and the gap does not close with sample size. On the process-reward artifact it is 0.881 against 0.864. We report the pair-weighted form because it is the term in (1); an author whose claim concerns a typical stratum should report the unweighted mean and say so.

H CONTROLLED-TEST PROTOCOL AND FULL DETECTOR TABLE

The datasets of Section 5 differ in domain, model, statistic and stratum count at once, so we also varied the stratification of one fixed dataset. Six uncertainty detectors score one generator's answers to 3,200 open-domain QA questions drawn equally, 400 each, from eight sources whose accuracies span 0.03 to 0.45 (TriviaQA and Natural Questions Open, validation splits; SciQ, WebQuestions, PopQA and SimpleQA, test splits; HotpotQA fullwiki, asked without its context, and TruthfulQA generation, validation splits; the first 400 questions of each after loading); merging the sources into four, two and one changes the number of strata while leaving every score, every outcome and the composition of the data untouched. Coarsening raises λ from 0.111 to 1.000, and both the largest departure of a permutation mean from 0.5 and the largest $A_{\text{pooled}} - A_{\text{within}}$ fall monotonically to zero with it, the single-stratum row recovering the degenerate case of (1) exactly. The merge changes the target stratum as well as λ, so the monotone pattern describes this dataset and is not a general effect of λ. The design also has a pre-specified primary outcome, fixed before these scores existed: that the detector ranked first by the pooled statistic would differ from the one ranked first within strata. It did not, at any granularity, so the outcome is negative. Table 12, the protocol, and the two pairs that do exchange places — neither pre-specified nor replicated across granularities — follow.

Table 12: The same 3,200 examples and six detectors at four stratification granularities. "Max departure" is the largest distance of a permutation mean from 0.5; "max gap" is the largest $A_{\text{pooled}} - A_{\text{within}}$.

strata	λ	null-reference range	max departure	max gap
8 sources	0.111	0.531–0.569	0.069	0.040
merged to 4	0.224	0.529–0.564	0.064	0.039
merged to 2	0.461	0.523–0.549	0.049	0.027
merged to 1	1.000	0.500–0.500	0.000	0.000

Pre-specification. The primary outcome, the secondary outcomes, the conditions requiring a rerun and the analysis code were fixed in a dated note before the eight-source scores existed, and the note is released with the artifact. It records the four-source result, which was already in hand, so that a reader can see what was known when the criterion was set. None of the rerun conditions was triggered: six or more sources were required and eight loaded, a per-source accuracy standard deviation of at least 0.05 was required and 0.140 was observed, and at least 90% of the intended 3,200 questions were required to complete, of which 3,200 did.

Generator and grading. One instruction-tuned model (`Qwen2.5-1.5B-Instruct` (Qwen Team, 2024)) answers each question with a short factual answer under a fixed prompt; correctness is normalised exact match against the released gold answers and their aliases. The answer is decoded greedily. Detector scores are computed from the same generation: the mean token log-probability and the minimum token probability of the answer, the mean per-token predictive entropy, p(true) as the model's normalised probability of the affirmative answer to whether its answer is correct (one extra forward pass), a string-match approximation to semantic entropy (the entropy over clusters of five additional samples at temperature one and top-p 0.95, clustered by normalised exact match rather than by meaning; the tables therefore call it string-cluster entropy, the code keeping the name `semantic_ent`), and the answer length in characters. Before any AUC is computed each score is oriented so that a higher value means more likely correct, which negates the two entropies and the length.

Table 13 gives every detector at the eight-source granularity. Two of the fifteen pairs are ordered differently by the two estimators, each with cluster-bootstrap support 0.32 for the disagreement; neither is the pre-specified comparison, and a different pair of pairs reversed in the four-source run, so we do not treat them as findings.

Table 13: All six uncertainty detectors at the eight-source granularity, with the rank each estimator assigns. Predictive entropy is first under both, which is the pre-specified outcome returning negative.

detector	pooled	rank	within	rank	null
pred_entropy	0.7392	1	0.7156	1	0.546
min_token_prob	0.7390	2	0.6990	3	0.569
mean_logprob	0.7357	3	0.7029	2	0.556
string-cluster ent.	0.6806	4	0.6428	5	0.559
p_true	0.6731	5	0.6502	4	0.536
length	0.5551	6	0.5372	6	0.531

I SURVEY OF REPORTING PRACTICE

Three artifacts do not say how common the convention is, so we coded the full text of two corpora, 210 papers for corpus 1 and 296 for corpus 2, storing the matched sentence behind every code. Corpus 1 is selection (process reward models, step-level verifiers, best-of-N reranking), where the use we code implies a within-problem comparison; corpus 2 is AUROC-native (hallucination detection, uncertainty quantification, confidence estimation, routing, error detection, guardrails), where a pooled AUROC is the reporting norm. Both corpora were fixed before coding and both are reported. For half of the confirmed exposed papers the conditional region of Section 4 is out of reach as well: recovering it needs the per-stratum outcome counts of the set actually evaluated, and where the outcomes are the paper's own generations judged by the paper's own judge those counts are released no more often than the scores are.

Table 14: Automated coding of 506 arXiv papers from full text. Corpus 1 is selection; corpus 2 is AUROC-native. Unresolved papers are excluded from every denominator rather than assigned an outcome. The 34 papers in the "not stratified" rows are the screen's flags; the manual reading of all 34 (text below) confirms 4 and leaves 2 borderline.

coding	corpus 1	corpus 2
no pooled pairwise statistic reported	190	174
pairwise, estimand unclear or single source	3	61
pairwise over heterogeneous strata, not stratified	6	28
pairwise over heterogeneous strata, stratified	11	24
unresolved (no HTML full text)	0	9

Across both corpora (Table 14) the automated screen initially classified 69 papers as reporting a pooled pairwise statistic over strata that differ, and 34 of them as giving no stratification cue. The contrast between corpora is what the scope rule predicts: the selection literature reports best-of-N accuracy, which is already within-problem, so the screen marks only 17 of its 210 papers as reporting such a statistic at all, against 52 of 296 in the AUROC-native corpus.

Manual re-adjudication of the flagged papers. The exposure code is the one that carries the claim, so every paper it flags was read in full, appendices and tables included, and coded on eight fields with a verbatim quote recorded for each: the statistic and where it appears; the pooling unit (one number over examples from several datasets, sources or problems, or one number per dataset); the estimand the paper's use implies; any stratified figure anywhere; release of per-example scores (the authors' own artifact, with the repository inspected where a link was given); whether the outcome labels are a public benchmark's or the paper's own generations judged; whether stratum keys are recoverable from public data; and a verdict. "Exposed" means that one pairwise statistic pools examples across heterogeneous strata and the paper's framing invites a within-stratum reading, whether or not the authors assert one. The coded files are released.

Of the 34, **6 are confirmed exposed** (three in each corpus; four clear and two borderline, Table 15), 21 compute each AUROC inside a single dataset and report a macro-average across datasets — the pattern reads "averaged over datasets" as pooling, and this one failure mode accounts for most of the screen's excess — 5 report a stratified figure somewhere, and 2 report no pairwise statistic (one uses

Table 15: The six papers a full reading returns as exposed: four clear, two borderline (marked). "Pool" is the set one pairwise statistic is computed over; "use" the reading the paper's own framing invites; "released" what is public; "computable" whether a reader can rebuild the design and compute the bounds of Section 4 from the printed figure.

arXiv id	pool	use	released	labels	computable
2505.12380	SQL-equivalence pairs across questions and databases	per-prompt group advantage	none	public	keys unclear
2510.04933	TruthfulQA subset plus synthetic pairs across domains	per-item detection	code only	mixed	no
2601.16397	sentence chunks across two clinical domains	per-summary flagging	promised, no link	judge-generated	no
2604.10660	steps across ProcessBench-MATH solutions	step-level ranking	code only	public	yes (Section 5)
2606.16999 (borderline)	candidates across the tasks of one benchmark cell	within-task reranking	manifests, no link	mixed	no
2607.12831 (borderline)	"contextual QA", composition unstated	global abstention	none	public	composition unknown

"AUC" for a cost–performance curve). Two of the 6 are borderline, both marked in Table 15: one does not state whether its single selective-prediction AUROC pools the suite's benchmarks, and one pools candidates in a diagnostic table rather than a headline.

Release: one of the 34 releases per-example scores, and it does not pool; of the 6 exposed papers four state no release or code only and two promise one without a link (linked repositories were inspected on 10 September 2026, and anonymised links that did not resolve are coded unverifiable). Labels: three of the 6 evaluate on public benchmarks whose outcome labels are public, two mix public labels with their own judged generations, and one labels its own generations with a judge. Public labels make the annotation-only interval computable for the benchmark as a whole, but both that interval and the bounds of Section 4 are properties of the pooled set actually evaluated, whose outcome counts and stratum memberships $\mathcal{I}_0$ requires; only one of the three states that set precisely enough to rebuild it (Table 15), and Section 5 computes that one.

Sampling. Papers were retrieved from the arXiv API, sorted by submission date descending, using five queries for corpus 1 (`"process reward model"`, `"step-level verifier"` OR `"step level verification"`, `"best-of-n"` AND `"verifier"`, `"reranking"` AND `"language model"` AND `"reward"`, `"outcome reward model"`) and six for corpus 2 (`"hallucination detection"` AND `"AUROC"`, `"uncertainty quantification"` AND `"large language model"`, `"confidence estimation"` AND `"language model"`, `"LLM routing"` OR `"query routing"`, `"error detection"` AND `"large language model"`, `"jailbreak detection"` OR `"guardrail"`). Results were deduplicated by arXiv id and truncated to the first 300 per corpus, of which 210 and 296 were retrieved. Full text came from the arXiv HTML rendering.

Coding. Every screening label comes from a deterministic regular expression, with the matched sentence stored, so each code is auditable. Two rules can only shrink the reported exposure: an unfetchable paper is never coded exposed, and any stratification cue moves a paper out of that category.