From boundary ties to tuned predictions: an implementation diagnosis of bootstrap feature aggregation

Abstract

Bootstrap feature aggregation combines complete rankings or top-k membership, but comparisons can conceal the conventions that turn tied scores into selected variables. We diagnose two established aggregators under shared resampling, explicit feature identities, and nested classifier tuning. Three public classification benchmarks and one correlated synthetic condition supplied 20 stratified bootstraps per training context. Mean average-rank and top-k frequency aggregation were compared under stored and permuted columns with index-based or original-ID priorities. Five-fold outer cross-validation repeated twice contained three-fold inner tuning; selection and scaling used training rows only. Rank-minus-frequency mean balanced-accuracy differences were positive for digits (0.009793–0.010341), negative for wine (−0.002222), zero for breast cancer, and tie-dependent for the synthetic condition (−0.008153–0.002378). All-feature prediction exceeded every reduced-feature arm on each dataset. A boundary-specific diagnosis separated harmless duplicate scores from unresolved top-k membership: no saved bootstrap cutoff was ambiguous, whereas final frequency aggregation admitted multiple subsets in three of ten synthetic refits. Only two changed under the tested permutation. In digits, two changed inner subsets switched regularization from C=10 to C=1; the final subset remained identical, but eight predicted labels changed. Original-ID priorities preserved selections, tuning choices, and probabilities across both orders. Synthetic selectors excluded all constructed noise yet repeatedly omitted one latent group. The results distinguish representation resolution, subset invariance, and complete-procedure invariance. This exploratory implementation study adds a traceable explanation of conditional differences, rather than a new aggregator or a universal predictive winner; four settings and dependent folds limit population inference.

Introduction

Feature selection serves several goals: prediction with fewer variables, reproducible variable choices, and interpretation of what the predictor uses (Guyon & Elisseeff, 2003; Saeys, Inza & Larrañaga, 2007). These goals need not coincide. A reduced subset can recur across training perturbations yet predict poorly; correlated variables can exchange positions without losing the same underlying information. Stability can refer to weights, ranks, or selected identities, so the object being assessed must be explicit (Kalousis, Prados & Hilario, 2007). This study concerns selected subsets and the predictions obtained after tuning on them.

Resampling aggregation offers an established response to unstable selections. Bagging motivated combining perturbed predictors (Breiman, 1996), and ensemble feature-selection research subsequently studied combining selector outputs (Saeys, Abeel & Van de Peer, 2008; Abeel et al., 2010). Homogeneous ensembles perturb one selector; heterogeneous ensembles combine different selectors (Seijo-Pardo et al., 2017; Bolón-Canedo & Alonso-Betanzos, 2019). Our design uses one univariate ANOVA-F selector across bootstrap samples. Its two representations retain different information: complete ranks preserve relative positions throughout the feature list, while top-k membership keeps only whether a variable passes a cutoff.

Both representations have direct precedents. Haury, Gestraud and Vert compared bootstrap ensemble-

mean rank aggregation and ensemble-stability top-k frequencies, alongside an exponential rank transformation, in molecular-signature selection (Haury, Gestraud & Vert, 2011). Salman and colleagues developed within- and between-selector aggregation (Salman et al., 2021) and then compared five score/rank aggregation rules for stability and prediction across twelve datasets (Salman, Alzaatreh & Sulieman, 2022). Their recent wrapper study extends bootstrap importance-score aggregation to regression-based selectors (Salman, Alzaatreh & Sulieman, 2026). Thus neither the two aggregators nor a comparison of their predictive and stability outcomes is new. Our question is narrower: which implementation decisions can change such a comparison, and through which stage do they reach the final prediction?

Ties make that question concrete. Frequency aggregates take integer values from zero to the bootstrap count. Mean ranks can also coincide. Many equal values may be harmless: tied features entirely inside or outside the selected subset cannot change membership. A tie spanning the kth boundary instead leaves several score-consistent subsets. A rule based on the current column index then gives storage order a role in selecting variables. The same feature identities, raw score arrays, and train/test splits can therefore produce different selections after a column permutation. This is an implementation perturbation, distinct from resampling the population. Documentation itself need not promise a portable tie order: scikit-learn's SelectKBest explicitly leaves equal-score ordering unspecified (scikit-learn developers). We use an explicit policy rather than attributing our observed ordering to that API.

Two further boundaries matter. First, a chance-adjusted subset-stability measure (Nogueira, Sechidis & Brown, 2018) does not establish predictive superiority or recovery of distinct information groups. Similar-feature-aware stability research addresses this interpretive issue (Bommert & Rahnenführer, 2020), while ranking evaluation over multiple cutoffs answers a broader question than fixed-k selection (Slavkov et al., 2020). Second, independent tuning evaluates a complete selection-and-classification procedure. Inner subset changes can alter the preferred regularization even if the final outer-training subset is unchanged. Keeping model selection separate from evaluation is therefore essential (Varma & Simon, 2006; Cawley & Talbot, 2010); checking only the final feature list cannot describe the full procedure.

We examine these boundaries using three public classification benchmarks and one constructed correlated condition, with shared bootstrap scores, nested logistic-regression tuning, single-F and all-feature baselines, and an original-ID priority control. The analysis distinguishes ties anywhere in an aggregate from unresolved top-k membership, follows an actual inner-selection-to-prediction trace, and separates exclusion of synthetic noise from coverage of distinct latent groups. Earlier small exploratory results motivated the design; changes in data conditions, splits, tuning, and invalid-score handling make this an exploratory extension rather than confirmation or a matched old-versus-new replication. The contribution is a reproducible conditional diagnosis of established procedures, with no claim of algorithm novelty or first discovery of tie sensitivity.

Relationship to broader ensemble studies

Published results give reasons to examine both benefits and failures. Ensemble selectors improved robustness in particular microarray settings (Abeel et al., 2010), whereas Haury et al. found that greater selection stability did not necessarily improve accuracy or interpretability (Haury, Gestraud & Vert, 2011). A text-classification study likewise reports ensemble combinations that underperform individual selectors (Parlak, 2023). Multi-domain stability analyses vary subset size and data conditions (Pes, 2020), and reliability-weighted aggregation is another established alternative (Ben Brahim & Limam, 2018). These findings motivate reporting contrary outcomes and the unreduced predictor, rather than interpreting an advantage over one reduced-feature comparator as evidence that feature selection itself is beneficial.

Recent work broadens the objective space to predictive performance, stability, redundancy, and computation (Barbieri, Grisci & Dorn, 2024), combines stable selection with application expertise (Xiang et al., 2024), and studies probability-based or multi-level ensembles in gene-expression and omics settings (Ahmadi & Mahmoodian, 2024; Ding et al., 2026). Our experiment does not benchmark those algorithms. It isolates the behavior of two familiar summaries under a modest shared budget. Table 1 makes the difference in scientific scope explicit.

Table 1. Scientific positioning relative to directly relevant work. Entries identify methodological scope, not a league table of empirical performance. The present algorithms are established; none of the unrun alternatives is an evaluated competitor.

Prior work	Main object studied	Relationship to the present comparison
Haury et al. (2011)	Mean-rank, top-k-frequency and exponential bootstrap ensembles; accuracy, stability and interpretation	Direct precedent for both aggregation representations; no first-comparison claim
Salman et al. (2021, 2022)	Within/between-selector frameworks and several score/rank aggregation rules	Broader aggregation comparisons; our one-selector, B=20 design is a diagnostic, not a reproduction
Slavkov et al. (2020)	Ranking evaluation through forward/reverse error curves	Addresses behavior across selection sizes; our fixed k cannot establish whole-ranking quality
Bommert and Rahnenführer (2020)	Stability measures accounting for similar features	Motivates identity-versus-information interpretation; our construction-defined groups are a different diagnostic
Present study	Boundary ambiguity, column-order controls, full nested-tuning trace, and group coverage	Identifies the stage and conditional propagation of observed implementation effects

Materials & Methods

Datasets and feature identities

We used the complete bundled datasets returned by scikit-learn 1.8.0 loaders `load_digits`, `load_wine`, and `load_breast_cancer`, alongside one custom synthetic dataset. Scikit-learn supplies the computational implementation (Pedregosa et al., 2011). Table 2 gives actual array dimensions and selected subset sizes. Digits contains 8×8 image-derived predictors and ten classes; wine contains 13 predictors and three classes. Breast cancer is treated solely as a binary classification benchmark. No clinical decision or utility is evaluated.

Table 2. Dataset sizes and fixed selection sizes. Class counts follow stored label order; n denotes observations, d features, and k selected features for all reduced-feature arms.

Dataset	n	d	Classes and counts	k
Digits	1,797	64	0–9: 178, 182, 177, 183, 181, 182, 181, 179, 174, 180	10
Wine	178	13	0–2: 59, 71, 48	5
Breast cancer	569	30	0–1: 212, 357	10
Correlated synthetic	600	60	0–1: 318, 282	10

For the synthetic condition, a NumPy default random generator seeded with 20261004 drew five independent standard-normal latent coordinates Z. Labels were Bernoulli draws with probability sigmoid(Zβ), where β = (1, −1, 0.8, −0.8, 0.6). The first five semantic predictors were the direct coordinates. Each coordinate supplied two proxies, $0.9Z_j + \sqrt{1 - 0.9^2}\varepsilon$, with independent standard-normal ε. Forty-five additional independent standard-normal columns formed the noise group. The same generator shuffled the completed 60-column array. This construction was implemented directly; `make_classification` was not its generator.

Original IDs 0–4 identify direct coordinates, 5–14 proxies, and 15–59 noise. Proxy pairs (5,6), (7,8), (9,10), (11,12), and (13,14) map to latent groups 0–4. Stored positions differ from these identities after shuffling. Benchmark original IDs are their initial column indices. All comparisons map selections back to original IDs. Latent coverage counts groups represented by either a direct coordinate or a proxy; proxies are associated predictors, not mislabeled noise or evidence of recovered causal variables.

Nested training and evaluation boundaries

Outer partitions were materialized once per dataset with `RepeatedStratifiedKFold`, five splits and two repeats, seed 20261004. For outer index o = 0,…,9, `StratifiedKFold` generated three shuffled inner folds with seed 20261004 + o + 1. All arms shared these partitions. Each inner-training context independently supplied feature scores, bootstrap aggregation, and scaling. Inner validation selected C, and the outer test partition remained excluded from these decisions. Selection and scaling were then recomputed on the complete outer-training partition before test prediction. This repeats feature selection within resampling rather than selecting once on pooled data, a distinction emphasized in work on selection bias and limited-sample validation (Ambroise & McLachlan, 2002; Vabalas et al., 2019). Figure 1 locates these boundaries.

Each arm searched $C \in \{0.1, 1, 10\}$, maximizing mean inner balanced accuracy. Exact objective ties selected the smallest C. Independent per-arm tuning means that the primary contrast compares complete procedures with equal search budgets. It does not hold regularization fixed to estimate an isolated selector effect. The structure follows the information separation described in the undated scikit-learn documentation *Nested versus non-nested cross-validation* (scikit-learn developers) and *Common pitfalls and recommended practices* (scikit-learn developers). Caching reused only an identical training context and configuration; it did not transfer a selector or scaler from a larger partition into an inner fold.

`StandardScaler` used its installed defaults: centering and unit-variance scaling, fitted on the exact train-

ing rows. Scaling was columnwise across the original array, then selected columns were extracted. Selection scores used raw training values before scaling. Logistic regression used `lbfgs`, maximum 2,000 iterations, tolerance 0.0001, an intercept, no class weights, no warm start, and random state 20261004. The installed default l1 ratio was 0.0, corresponding to L2 regularization; multiclass handling used the installed supported default rather than an explicitly passed `multi_class` option. Every classifier received columns sorted by original ID, removing solver input order as a competing explanation for a storage-order effect.

The recorded environment was Python 3.12.4, NumPy 2.2.6, SciPy 1.17.1, and scikit-learn 1.8.0 on Windows, with one computational thread. Documentation accessed for the design was a later stable snapshot; that snapshot is not the executed software version.

Computational and writing assistance

OpenAI Codex and delegated GPT-6.1 Sol agents assisted with literature organization, writing, analysis code, and figure code to support the exploratory investigation. The recorded requested model identifier was `gpt-6.1-sol`; Codex CLI 0.160.0 was used for the experimental agent workflow. The service did not expose a more specific resolved model build. This disclosure describes assistance, not authorship. Numerical claims were checked against retained arrays and independent reconstruction. The plotted scientific content was rendered deterministically from those arrays using Matplotlib 3.10.9, rather than generated as an illustrative bitmap. AI-assisted drafts and subsequent revisions are retained with the editable sources.

Shared bootstrap scores and aggregation

Each exact training context generated B = 20 stratified bootstrap samples with replacement. Within each class, sampling retained the training class count, giving the original context size. Samples were shared across representations, column orders, and tie modes. Deterministic bootstrap seeds were the little-endian integer from the first four SHA256 bytes of `20261004|context|bootstrap_b`, with zero-based b. Context names encode dataset, outer index, and inner index or refit stage. A separate `priority` purpose in the same seed derivation generated one original-ID priority permutation per context.

The score matrix contained raw `f_classif` ANOVA F statistics, not ANOVA p-values or classifier coefficients. A feature was constant when its range across every row of the exact score-fitting sample was zero. Constants, NaNs, and negative infinities were assigned effective score $-\infty$; finite nonconstant scores and nonconstant positive infinities were retained. Each vector required at least k valid features. Unsanitized scores and masks remained available. These rules specify behavior without assuming that every nonfinite case occurred.

For bootstrap $b = 1, \ldots, B$ and feature $j = 1, \ldots, d$, average rank R_{bj} was computed by SciPy `rankdata` on negative effective scores, using `method='average'`. Mathematical indices enumerate bootstrap rows and original-ID feature columns; the saved bootstrap indices and original IDs are zero-based. Rank 1 is best. Equal scores receive the average of their occupied ranks, independently of lexical or random priority. This is the documented average-rank convention, distinct from ordinal ranks that assign occurrence-dependent consecutive positions (SciPy developers). The mean rank of feature j is

$$\overline{R}_j = \frac{1}{B} \sum_{b=1}^{B} R_{bj}, \qquad B = 20.$$

Mean-rank aggregation selects the k smallest $\overline{R}_j$ values; the implementation stores $-\overline{R}_j$ so larger aggregate values are better. Only ties in final mean ranks use the specified priority policy.

Frequency aggregation first orders each effective score vector, resolving exact ties with that policy. Let T_b denote its selected top-k feature set, and let $\mathbf{1}\{\cdot\}$ denote an indicator equal to one when its condition holds and zero otherwise. Membership and the vote count for feature j are

$$V_{bj} = \mathbf{1}\{j \in T_b\}, \qquad W_j = \sum_{b=1}^{B} V_{bj}.$$

It selects the k largest W_j values, resolving final count ties with the same policy. Thus frequency priorities operate both within bootstrap membership selection and at final aggregation. Equality is exact floating-point equality, without a numerical tolerance.

The two column conditions were stored order and one fixed permutation generated with seed 20261005 for each dimensionality. Lexical ties prefer the current column index. Random ties prefer a context-specific permutation of original IDs, shared across methods, both orders, every bootstrap, and final aggregation. Random priority is an implementation control for index-order invariance; it is not a population perturbation or false-selection error-control procedure.

The factorial comparison produced eight aggregation arms: two representations × two orders × two tie policies. Two additional arms applied a single F vector fitted on the context's full training rows, using lexical ties under each order. One all-feature baseline bypassed selection. The resulting 11 arms used identical evaluation boundaries and independent C searches.

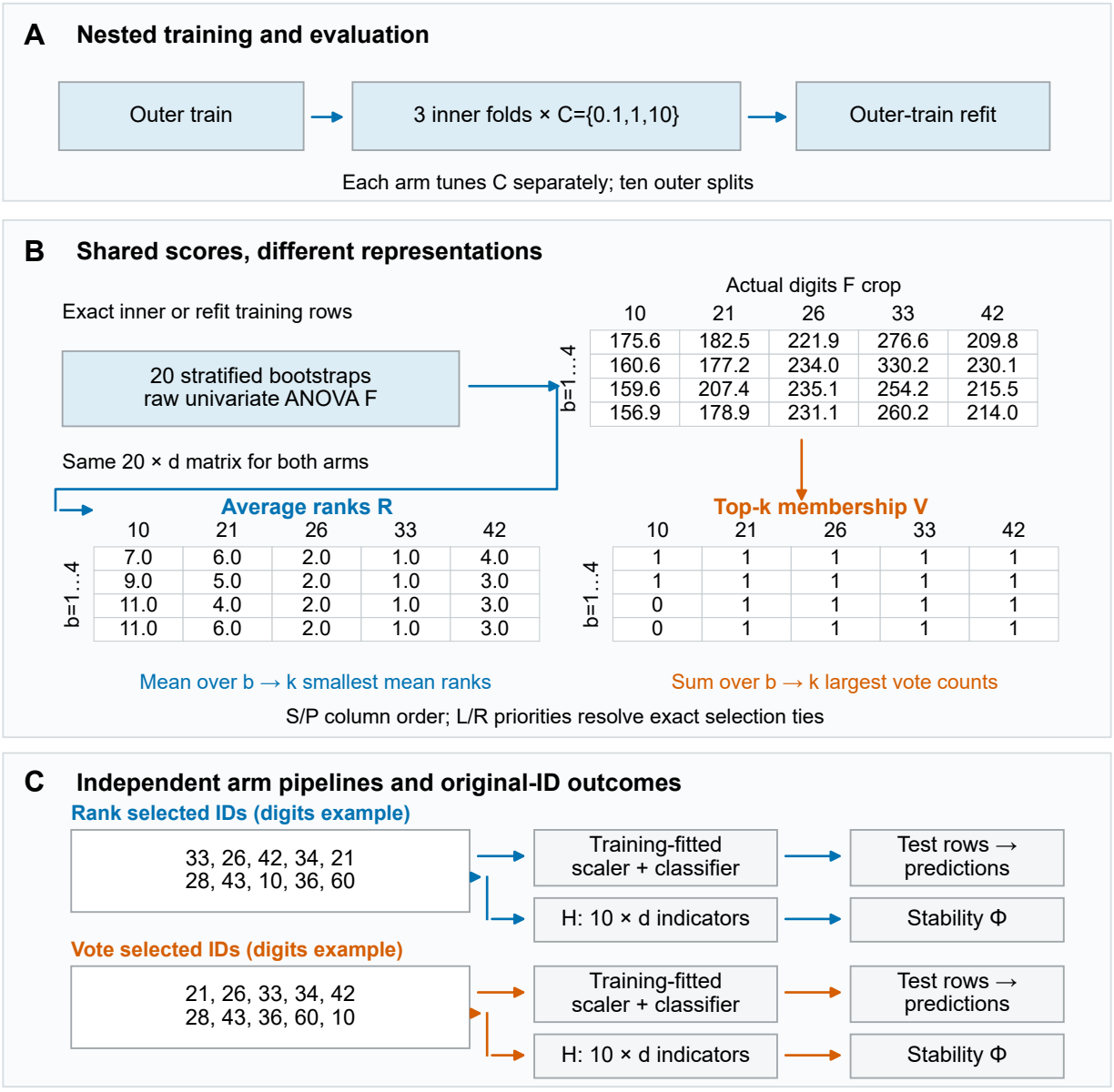


Figure 1. Nested evaluation and the two bootstrap representations. (A) Each of ten outer splits (five folds, two repeats) contains three inner training/validation folds. Regularization C is chosen independently for each arm, followed by an outer-training refit and held-out prediction. Selection, scaling, and fitting use the applicable training partition. (B) Each exact training context supplies a shared 20×d ANOVA-F matrix. Average ranks retain within-row ordering, with average ranks for exact ties; frequency records top-k membership and sums votes. Displayed 4×5 crops use bootstrap rows 1–4 and original IDs 10, 21, 26, 33, and 42 from digits context main__digits__outer0__refit. Transformations use all 64 features; aggregation and selection use all 20 bootstraps. (C) Separate rank and vote paths list their complete ordered top-ten original IDs for that context, then fit independently. No feature-set union is performed. Classifier columns are sorted by original ID. The separate selected-set indicator matrix H contains all ten outer refits and d original-ID columns for each arm; it yields stability Φ and does not feed classifier training. S/P denote stored/permuted columns; L/R denote current-column lexical/original-ID-keyed random priorities. Random priorities resolve final mean-rank ties and both bootstrap and final frequency ties. No held-out label enters selection or scaling. The diagram explains the implemented procedure with actual intermediate arrays; it is not causal evidence.

Outcomes and descriptive diagnostics

Balanced accuracy, the mean of class recalls, was primary; macro F1 was secondary. Balanced accuracy gives classes equal weight rather than weighting recall by observed prevalence (Brodersen et al., 2010). We paired rank and frequency outcomes within dataset, outer split, order, and tie policy. Means, observed ranges, and positive/zero/negative paired counts describe those ten matches. Overlapping training partitions make folds correlated; no naive fold confidence intervals or p-values were calculated. Four chosen datasets also do not constitute independent population replications or justify a cross-dataset inferential winner (Demšar, 2006).

For each arm separately, let H be its $M \times d$ original-ID selection indicator matrix, with $H_{mj} = 1$ when outer refit m selects feature j and zero otherwise ($m = 1, \ldots, M$, $j = 1, \ldots, d$). This is the selected-set matrix H in Figure 1C, distinct from the synthetic latent coordinates. Feature inclusion proportion p_j, mean selected proportion q, and unbiased sample variance s_j^2 are

$$p_j = \frac{1}{M}\sum_{m=1}^{M} H_{mj}, \qquad q = \frac{1}{d}\sum_{j=1}^{d} p_j,$$

$$s_j^2 = \frac{1}{M-1}\sum_{m=1}^{M}(H_{mj} - p_j)^2 = \frac{M}{M-1}p_j(1-p_j).$$

We calculated Nogueira stability (Nogueira, Sechidis & Brown, 2018) as

$$\Phi = 1 - \frac{\frac{1}{d}\sum_{j=1}^{d} s_j^2}{q(1-q)}, \qquad M > 1, \quad 0 < q < 1.$$

Actual M, d, and selection cardinalities determine the calculation; here each complete arm had $M = 10$. Negative values would be retained. At no-feature ($q = 0$) or all-feature ($q = 1$) selection the denominator is zero, so Φ is undefined; it is also undefined for $M \leq 1$. Mean Jaccard similarity uses all 45 outer-set pairs; its all-feature value is 1. Stability includes changes from both training splits and context-specific priority randomness.

Boundary ambiguity and complete-procedure diagnostics

We distinguished duplicate aggregate values from ties that straddle the selection cutoff. `final_tie_excess` equals d minus the number of unique aggregate values across the entire vector. For a fixed aggregate vector, let t be the kth-largest value, a the number strictly above t, and e the number equal to t. The remaining r=k−a places must be chosen from that tied block. The number of distinct score-consistent top-k subsets is

$$\mathcal{A} = \binom{e}{r}, \qquad r = k - a, \quad 1 \leq r \leq e.$$

Every permitted subset contains the a strictly higher features and r of the e tied features, which gives this elementary counting identity. Membership is ambiguous precisely when e>r. A tied block wholly included at the boundary (e=r) has $\mathcal{A} = 1$, as do vectors with a unique kth value. The count describes a fixed vector; it is not a probability, a prediction bound, or a claim that coupled bootstrap/final priorities can attain all those subsets.

Frequency's whole-vector duplicate excess also has a simple resolution bound. Because each vote count belongs to $\{0, \ldots, B\}$, at most B+1 distinct values are available. Thus

$$d - |\{W_1, \ldots, W_d\}| \geq \max\{0, d - (B + 1)\}.$$

When d>B+1, duplicates are inevitable somewhere in the vector. This pigeonhole bound does not locate them at the kth boundary and therefore does not imply an ambiguous selected set.

Cutoff diagnostics were calculated for every saved bootstrap vector and final aggregate. We examined 160 exact training contexts: four datasets × ten outer splits × three inner fits plus one refit. There were 1,600 reduced-arm final vectors and 12,800 frequency-bootstrap order/priority vectors. The latter repeat 3,200 unique raw bootstrap score rows across four implementation conditions; they are not independent samples. These explanations were chosen after the main results were available and are exploratory analyses of saved evidence. They introduce no new fits or fresh validation data.

Column-order comparisons examined inner selections at each matched C and outer refits at independently chosen C, with probabilities aligned by held row and class. For every digits outer split we retained complete three-C objective vectors, selected identities, chosen C, and predictions for both methods and priority rules. This permits a trace through the actual tuning decision, including unchanged inner folds and original-ID controls, rather than attributing a final prediction difference solely to the final feature list.

For the synthetic condition, group g was covered in refit m when at least one direct coordinate or proxy belonging to g was selected. A five-column group indicator matrix supplies group inclusion frequencies, coverage, and descriptive stability. Group Φ uses the same finite-M formula above with d=5 and the actual mean covered-group proportion; group cardinality can vary across refits. All-five coverage has an undefined denominator. Feature and group Φ use different universes and chance baselines and are not a common scale for claiming a stability increase or decrease. Group frequencies, rather than causal interpretation of selected proxies, are the principal coverage result.

Results

Conditional prediction contrasts and baselines

Mean-rank aggregation did not uniformly outperform frequency aggregation. Table 3 reports the complete paired contrasts, and Figure 2A displays their constituent observations. In digits, rank improved balanced accuracy in three splits and tied in seven under every condition. The mean gain was 0.009793 for stored lexical ties and 0.010341 otherwise. Wine showed a loss in one split and ties in nine, giving −0.002222 in all four conditions. Breast cancer showed ten zero differences throughout. These zeros are observed agreement under this design, not a statistical equivalence result.

Table 3. Rank-minus-frequency differences across ten matched outer splits. S/P denote stored/permuted order; L/R denote lexical/original-ID random priority. Values are proportions, so 0.01 equals one percentage point. Signs count positive/zero/negative balanced-accuracy differences.

Dataset	Condition	Mean Δ balanced accuracy	Mean Δ macro F1	Signs
Digits	S/L	0.009793	0.009995	3/7/0
Digits	P/L, S/R, P/R	0.010341	0.010659	3/7/0 each
Wine	All four	−0.002222	−0.002986	0/9/1 each
Breast cancer	All four	0.000000	0.000000	0/10/0 each
Correlated synthetic	S/L	−0.008153	−0.008211	1/3/6
Correlated synthetic	P/L	0.001894	0.001778	4/3/3
Correlated synthetic	S/R, P/R	0.002378	0.001925	4/3/3 each

Synthetic results changed direction: stored lexical frequency averaged 0.721885 balanced accuracy versus rank's 0.713732. Permuted lexical frequency averaged 0.711838, and random-priority frequency 0.711354 under either order. Paired synthetic differences ranged from −0.049107 to 0.025670 for S/L, −0.033482 to 0.025670 for P/L, and −0.023810 to 0.025670 under random priorities. The corresponding macro-F1 directions agreed with balanced accuracy. Rank's digits gains reached 0.038973 for S/L and 0.044453 otherwise; wine's single negative difference was −0.022222.

All-feature prediction supplied an essential reference point (Figure 2B). Its mean balanced accuracy exceeded every reduced-feature arm in all four datasets (Table 4). Rank improved on single-F selection in digits and wine but fell below it in the synthetic condition. Both aggregators slightly exceeded single F for breast cancer. Frequency fell below single F in digits and exceeded it in wine. These comparisons show that choosing an aggregator and choosing whether to reduce features answer different questions.

Table 4. Mean balanced accuracy / macro F1 for stored lexical arms and baselines, each over ten outer splits. The single-F values also apply to permuted order. Other frequency conditions are specified in Table 3 and Results; rank values are unchanged across conditions.

Dataset	Average rank	Frequency	Single F	All features
Digits	0.847746 / 0.846706	0.837953 / 0.836711	0.840019 / 0.838592	0.968281 / 0.968389
Wine	0.972857 / 0.969976	0.975079 / 0.972962	0.971376 / 0.970360	0.982049 / 0.980049
Breast cancer	0.954834 / 0.956495	0.954834 / 0.956495	0.953671 / 0.955539	0.971998 / 0.974415
Correlated synthetic	0.713732 / 0.712739	0.721885 / 0.720950	0.715316 / 0.714306	0.730898 / 0.730408

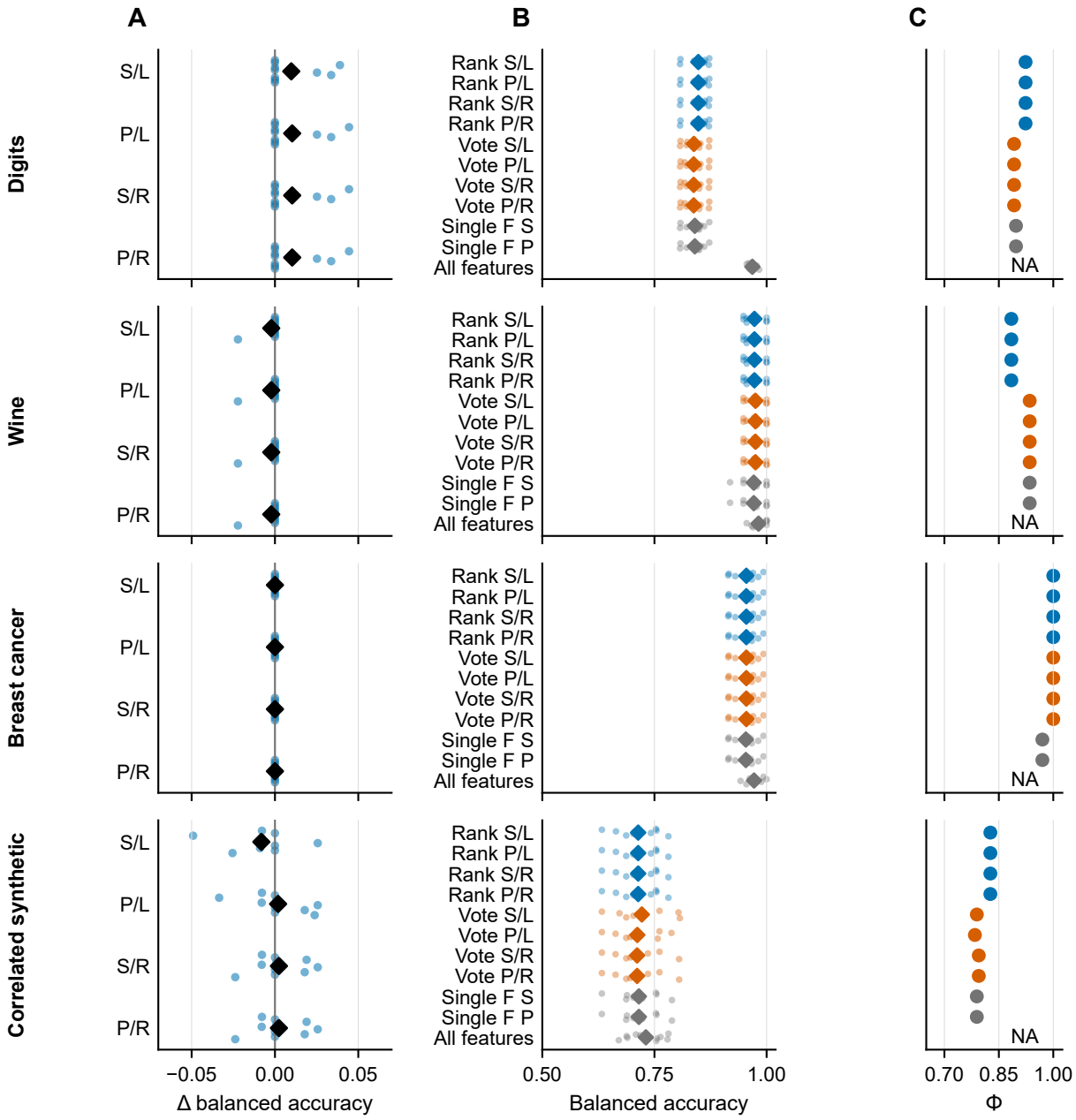


Figure 2. Paired predictive differences and full baseline comparisons. Rows correspond to the four fixed datasets; aligned columns use common x scales displayed on the bottom row. (A) Ten matched outer balanced-accuracy differences (average rank minus frequency) per column/tie condition; light points are individual paired observations, diamonds their arithmetic means, and the vertical line is zero. (B) All 11 arms, including single-F selection and all-feature baselines: points denote individual outer balanced accuracies and diamonds their means. (C) Nogueira Φ from the ten full outer-refit selected sets for each arm; all-feature Φ is undefined and is shown as NA. S/P denote stored/permuted columns; L/R denote lexical/original-ID-keyed random priorities. The folds overlap and are correlated; no error bars, independent-fold confidence intervals or hypothesis tests are implied. Each arm tunes C independently. Breast cancer is a classification benchmark; differences here establish no clinical performance claim.

Stability does not select a universal predictive winner

Digits rank stability was $\Phi = 0.923621$, compared with frequency's 0.892016 and single F's 0.897284. Their mean Jaccard similarities were 0.882828, 0.839057, and 0.844444. Wine reversed this pattern: frequency and single F both had $\Phi = 0.935000$ and Jaccard = 0.933333, versus rank's $\Phi = 0.884444$ and Jaccard = 0.881481. Breast-cancer aggregation sets were identical across all refits, with Φ and Jaccard both 1; single F had $\Phi = 0.970000$ and Jaccard = 0.963636.

Synthetic rank stability was $\Phi = 0.826667$ and Jaccard = 0.753743. Frequency Φ varied from 0.784000 under P/L to 0.794667 under random priorities, with S/L at 0.789333; Jaccard ranged from 0.704577 to 0.717682. Single F had $\Phi = 0.789333$ and Jaccard = 0.708521. Rank was more stable in this condition, yet S/L frequency predicted better on average. Figure 2C therefore complements predictive results rather than supplying a substitute ranking of predictive methods. All-feature Φ remains undefined throughout.

Ties, column order, and complete tuning

Figure 3 summarizes the observed order differences; Figure 4 separates their selection and tuning stages. Within-bootstrap duplicate-score excess averaged 3.91125 in digits, ranging from 2 to 7 across 800 vectors. The other datasets had no exact within-bootstrap score ties. Final frequency aggregates nevertheless contained many duplicates: mean refit `final_tie_excess` was 56.0 for digits, 7.6 for wine, 25.8 for breast cancer, and 48.5 for the synthetic condition. Corresponding rank values were 3.1, 0.2, 0.9, and 2.9. These are whole-vector counts; numerous equal zero votes below the cutoff can increase them without changing selected membership.

None of the 12,800 saved frequency-bootstrap order/priority vectors had an ambiguous top-k boundary. Thus the within-bootstrap membership sets were unique for these executions, including digits vectors with tied invalid features below the cutoff. The observed frequency order effects arose at final vote aggregation, rather than through changed bootstrap membership. This local result does not imply that bootstrap score ties are harmless in general.

The final boundary census is shown in Table 5 and Figure 4A. Rank had an ambiguous cutoff in one digits and one wine inner context, and none in any outer refit. Frequency had ambiguous boundaries in 3/30 digits, 2/30 wine, 1/30 breast-cancer, and 7/30 synthetic inner contexts. Single-F boundaries were unambiguous. Only synthetic frequency outer refits admitted multiple subsets: three of ten, allowing three, three, and five score-consistent subsets at zero-based outer indices 2, 3, and 9. Two actually changed under the tested permutation. Ambiguity therefore permits order sensitivity without guaranteeing a change under any one permutation.

Table 5. Boundary ambiguity in stored-order lexical aggregates. Counts refer to distinct exact training contexts; maxima count score-consistent top-k subsets of a fixed vector, not predictions or independent repetitions. All other conditions and negative cases are retained in the diagnostic data.

Dataset	Rank inner /30	Vote inner /30	Vote max inner subsets	Rank refit /10	Vote refit /10	Vote max refit subsets
Digits	1	3	2	0	0	1
Wine	1	2	3	0	0	1
Breast cancer	0	1	2	0	0	1
Correlated synthetic	0	7	4	0	3	5

Lexical frequency changed two synthetic outer-refit sets under permutation, averaging Jaccard 0.963636 between matched orders (Figure 3A). Three chosen C values changed; mean permuted-minus-stored balanced accuracy was −0.010047, and the maximum absolute outer difference was 0.049290. Its five changed inner sets reinforce that the affected object extends beyond outer membership. Digits lexical frequency changed three inner sets, but no outer-refit sets. Nevertheless, one outer C changed, maximum probability discrepancy was 0.284696, and maximum absolute balanced-accuracy change was 0.005480 (Figure 3B). Digits rank changed one inner set without changing chosen C or outer predictions. Breast-cancer frequency changed one inner set but left all outer outcomes unchanged; wine had no order changes.

For digits rank, C = 1 was chosen three times and C = 10 seven times. Stored lexical frequency chose these values two and eight times, respectively; its permuted and random variants matched rank's counts. Synthetic rank selected C = 0.1, 1, 10 in 4, 5, 1 refits; frequency S/L counts were 4, 4, 2, compared with 5, 4, 1 for P/L and random priorities. Counts summarize tuning decisions without implying that C alone explains performance.

The complete digits trace identifies how a final-list check can miss a predictive change (Figure 4B). In outer split 2, lexical frequency replaced original feature 10 with 30 in inner folds 0 and 1; inner fold 2 remained unchanged. The stored mean inner balanced accuracies at C=0.1, 1, and 10 were 0.836078, 0.847108, and 0.847239. The permuted means were 0.820202, 0.825066, and 0.823637. Consequently C changed from 10 to 1. The stored winning margin over C=1 was only 0.000132; the permuted margin over C=10 was 0.001429. These are observed tuning-criterion margins, not population effect sizes.

Both final refits selected the same ten original IDs, but their tuned predictors changed eight labels among the same 359 held observations. Balanced accuracy changed from 0.823336 to 0.817856, with maximum absolute probability discrepancy 0.284696. The same-outer original-ID control preserved all inner sets, objective vectors, selected C, and probabilities. The unchanged lexical inner fold also retained identical objectives at every C. Other changed inner sets did not switch C, so an inner selection change is not sufficient for a tuning change. This trace locates an actual propagation path; it does not assign a causal fraction to either changed fold or report an unrun fixed-C counterfactual.

The random original-ID control left selected sets unchanged in all 320 context/method order comparisons. All 800 saved probability comparisons had exactly zero discrepancy, and matched outer C choices agreed. Sorted original-ID classifier columns make this control interpretable: storage permutation did not alter classifier input ordering. The result establishes invariance for these executions, not general robustness to population change.

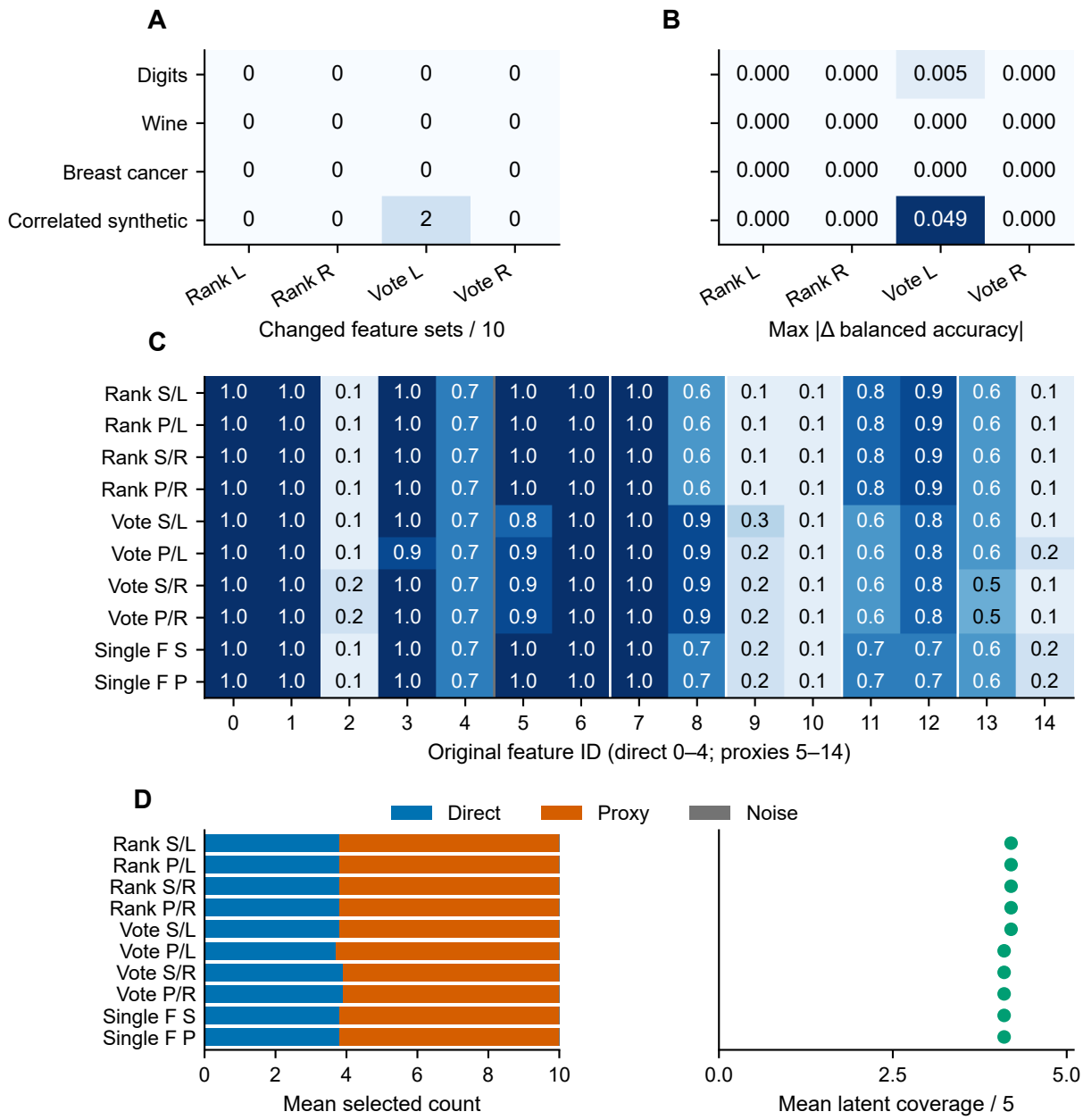


Figure 3. Column-order control and the identity of selected synthetic predictors. (A) Number of outer refits whose selected original-ID sets changed under the fixed column permutation (out of ten), by dataset, representation and tie policy. (B) Maximum absolute change in outer balanced accuracy for the same stored-versus-permuted comparisons. A zero outer-refit set change does not require zero predictive change, because inner selections and independently tuned C also belong to the compared procedures. Original-ID-keyed priorities are a code control; their zero differences do not establish independent robustness or error control. (C) Synthetic feature inclusion frequency over ten outer refits for all ten selected-feature arms; columns are the five direct latent features and ten proxies grouped by their parent latent factor, preserving original IDs: proxy pairs (5,6), (7,8), (9,10), (11,12), (13,14) correspond to direct factors 0–4. Noise-feature frequencies are all zero for these ten arms and are reported explicitly rather than omitted without explanation. (D) Mean selected counts of direct, proxy and noise features for the same arms, and mean number of distinct latent factors covered (out of five). The all-feature baseline selects 5 direct, 10 proxy and 45 noise columns and covers all five factors; it is not a fixed-k selector and is reported in the caption rather than forced onto the ten-feature count scale. Group labels come from the frozen synthetic construction, not recovered causal truth. Stability includes both split and priority-RNG variation.

Synthetic identities and score warnings

All ten reduced-feature arms selected zero of the 45 constructed noise features in every synthetic refit. Rank selected a mean 3.8 direct and 6.2 proxy features and covered 4.2 latent groups. Frequency S/L had the same mean counts and coverage; P/L selected 3.7 direct and 6.3 proxies, while random variants selected 3.9 direct and 6.1 proxies. Those latter frequency conditions covered 4.1 groups. Single F selected 3.8 direct and 6.2 proxies, covering 4.1 groups (Figure 3C–D).

Identity-level frequencies reveal incomplete representation despite noise exclusion. Rank always selected direct IDs 0, 1, and 3, selected ID 2 in only one refit, and ID 4 in seven. Proxies 5, 6, and 7 were always selected; inclusion of other proxies varied. Selecting multiple predictors from a strongly represented group can leave another group unrepresented. The full-feature baseline includes all five direct coordinates, ten proxies, and 45 noise columns and covers all five groups. Its better mean prediction does not identify which of those additional columns accounts for the difference.

The group representation makes the omissions explicit (Figure 4C). All reduced arms covered groups 0, 1, and 3 in every outer refit. Rank covered group 2 in only 3/10 and group 4 in 9/10. Frequency S/L covered these groups in 4/10 and 8/10; P/L in 3/10 and 8/10; both random-priority variants in 4/10 and 7/10. Single F covered them in 3/10 and 8/10. Within the five-group representation, descriptive Φ was 0.503968 for rank, 0.338624 for frequency S/L, 0.442939 for P/L, 0.322493 for random frequency, and 0.442939 for single F. These values have a different null denominator from feature-level Φ and are not evidence of a same-scale stability decrease. The substantive result is persistent incomplete coverage despite complete noise exclusion.

Raw ANOVA scoring retained 3,765 NaNs, all matching constant-column masks in digits. The 40 digits contexts, each with 20 bootstrap vectors and one single-F vector, generated 840 constant-feature warnings and 840 invalid-division warnings. All contexts retained at least k valid predictors. The other three datasets had no observed nonfinite scores or corresponding warnings. No positive or negative infinities were observed. Consequently, positive-infinity retention is a specified policy whose behavior was not empirically exercised here.

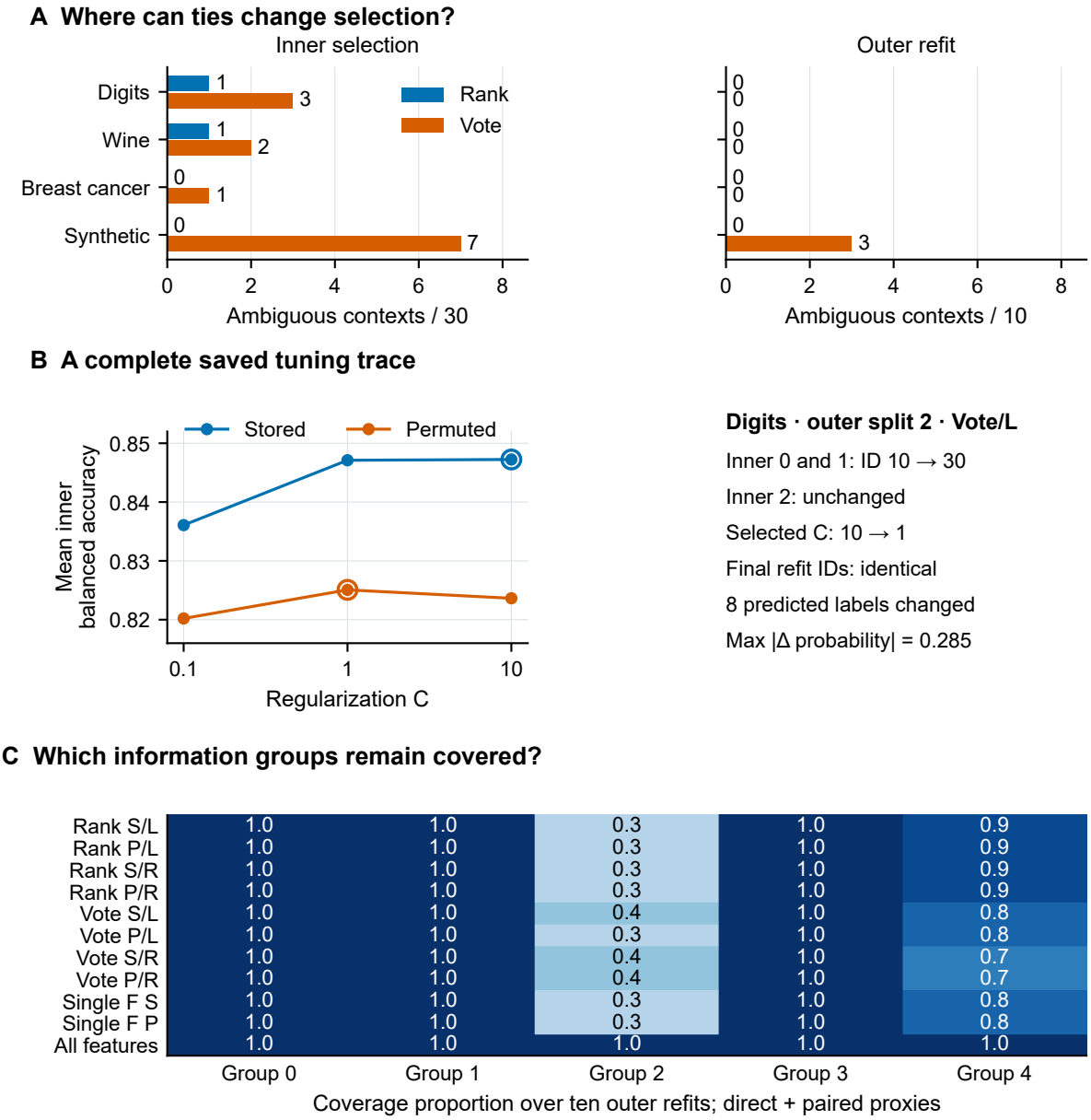


Figure 4. Boundary ambiguity, complete tuning, and construction-defined group coverage. (A) Ambiguous final cutoffs under stored lexical priorities, separately across 30 inner and ten refit contexts per dataset. Rank and vote counts refer to distinct vectors for their respective procedures, not independent experiments. None of the saved within-bootstrap frequency cutoffs was ambiguous. (B) Digits outer split 2: mean inner balanced accuracy at all three C values for stored and permuted lexical-frequency procedures; circles mark the selected C. Inner folds 0 and 1 exchange feature 10 for 30; the third remains unchanged. The final ten-feature sets agree, but eight predicted labels among 359 held observations differ after C changes from 10 to 1. The maximum absolute probability discrepancy is 0.284696. The original-ID control preserves objectives, C, and predictions. (C) Group-coverage frequencies across ten synthetic outer refits for all 11 arms. Direct IDs 0–4 and their respective proxy pairs define five information groups. Frequency P/L and random-priority conditions retain distinct coverage values. All-feature coverage is one for every group. Construction labels do not establish causal recovery, and coverage frequencies are descriptive under overlapping folds.

Discussion

Three objects that a performance comparison can conceal

The results distinguish representation resolution, selected-subset invariance, and complete-procedure invariance. Average-rank aggregation predicted better than frequency in digits, slightly worse in wine, identically in breast cancer, and conditionally in the correlated construction. The all-feature predictor had the highest mean balanced accuracy in every setting. The central implication is therefore not a preferred aggregator: it is that the evaluated procedure includes information representation, cutoff resolution, and inner tuning. An observed advantage can depend on a convention whose scientific role is easily overlooked.

The boundary diagnosis explains why global tie prevalence is a weak proxy for order sensitivity. With B=20, vote counts have only 21 possible values. Each synthetic vote vector therefore has at least 39 duplicate excess entries; the observed mean refit excess was 48.5. Yet only three of ten refits had unresolved membership. Zero-vote blocks below the cutoff contribute many duplicates while leaving the selected set fixed. The exact admissible-subset count separates this limited representation resolution from a decision boundary that permits arbitrary feature replacement.

The saved bootstrap cutoffs sharpen the explanation further. All were unique here, so changing the lexical priority could not change within-bootstrap membership or its vote counts. Order effects arose when the votes were combined and the final boundary resolved. Three ambiguous synthetic refits yielded only two observed set changes: a particular permutation can preserve the chosen members of a tied block. Wine had no observed order change; breast-cancer frequency changed one inner set but retained C and outer predictions. The useful condition is permission for sensitivity, rather than a claim that every boundary tie must change the result.

How inner selection reaches a tuned prediction

The digits trace demonstrates the additional boundary introduced by tuning. Two inner folds exchanged feature 10 for 30; their three-C validation objectives changed, and the winning regularization switched. The eventual refit selected the same columns in both orders. Its probability and label differences were therefore invisible to a final-feature-list audit. Independent tuning asks which complete procedure predicts better; it cannot be interpreted as an isolated effect of the final selector.

The stored C=10 objective exceeded C=1 by only 0.000132. This small observed margin makes the switch understandable without supplying a sampling probability for such events. The unchanged inner fold and original-ID controls retain the corresponding objectives and predictions, while other inner selection changes leave the winning C fixed. These negative cases identify where a perturbation stopped propagating. They strengthen the explanation without an unrun decomposition into separate causal contributions from each fold. The finite-sample sensitivity of model-selection criteria is established more generally by Cawley and Talbot (Cawley & Talbot, 2010); our trace is a concrete instance under this pipeline.

Original-ID priorities achieved exactly matching selections, chosen C, and saved probabilities across the two tested column orders. For a fixed score vector, a priority attached to feature identity moves with the feature, whereas an index-based priority changes when columns move. Sorting classifier inputs by original ID then removes solver column order as a competing explanation. This makes the control interpretable, but its value is reproducible implementation behavior. Synthetic random-priority frequency predicted worse than stored lexical frequency, so eliminating an arbitrary index dependency did not confer a predictive benefit in these results.

Established aggregation methods, different questions

Haury et al. already studied mean-rank and selection-frequency bootstrap ensembles and found that accuracy, stability, and interpretation could disagree (Haury, Gestraud & Vert, 2011). Our observed mixture of positive, zero, and negative comparisons is consistent with the need to report all three objects separately. Salman et al. compared importance-score and rank-based aggregation under broader selector and dataset choices (Salman et al., 2021; Salman, Alzaatreh & Sulieman, 2022). Their importance-score averages are not the mean of average ranks studied here. Rank transformation removes score magnitude; top-k membership further removes positions outside the selected group and differences among its members. The different information retained supplies a plausible explanation, rather than a causal proof that retaining more information improves classification.

Other rank aggregators, such as robust rank aggregation, use their own agreement or null-model constructions (Kolde et al., 2012). Recent wrapper aggregation explores scoring under collinearity (Salman, Alzaatreh & Sulieman, 2026), and multi-level ensembles combine several forms of selection in a different application setting (Ding et al., 2026). Those methods answer broader design questions and were not run in this comparison. The incremental contribution here is locating an observed implementation effect and following its entire saved tuning path. A transparent small experiment can expose that behavior; it cannot replace the coverage of a broad multi-method benchmark.

The terminology of stability selection also requires care. Top-k bootstrap frequency is a counting rule. Stability selection, complementary-pairs stability selection, and integrated path stability selection connect selection frequencies or paths to specified error-control methods (Meinshausen & Bühlmann, 2010; Shah & Samworth, 2013; Melikechi & Miller, 2026). Our original-ID priority control supplies none of those guarantees. The algorithm and conditions, rather than a similar label or the presence of resampling, determine which conclusions transfer.

Stability, information coverage, and the unreduced baseline

Chance-adjusted subset stability correctly retains the selected proportion and finite-M variance factor (Nogueira, Sechidis & Brown, 2018). It still requires scientific interpretation: synthetic rank was more stable at the identifier level, while stored lexical frequency predicted better. Published ensemble studies likewise provide both positive robustness results (Abeel et al., 2010) and contrary domain-specific outcomes (Haury, Gestraud & Vert, 2011; Parlak, 2023). Maximizing agreement alone can favor repeated selection of a limited information set.

The synthetic groups make this distinction visible. Excluding every noise feature coexisted with frequent omission of group 2, even though ten variables were selected. Multiple direct/proxy alternatives represented already-covered groups. Similar-feature-aware stability measures address how identifier exchanges can conceal retained information (Bommert & Rahnenführer, 2020); our construction-defined coverage asks the complementary question of which distinct sources remain absent. Proxies carry information and are not false positives. Group-level frequencies and their own stability denominator expose the observed omissions without converting them into causal recovery claims. Rank-sensitive stability is another object altogether (Rakesh, Anwit & Jana, 2023).

The all-feature predictor prevents a reduced-method comparison from becoming the whole argument. In digits, keeping ten of 64 predictors was a severe compression, and all features yielded mean balanced accuracy 0.968281 versus rank's 0.847746. This supports retaining the unreduced baseline in interpreting the fixed-k result. For the synthetic construction, better all-feature prediction does not reveal whether omitted

groups, additional proxies, regularization, or other columns produced the difference. Error curves across selection sizes (Slavkov et al., 2020) would address that question more fully than our single cutoff. Benchmarks that consider performance, redundancy, stability, and computation together (Barbieri, Grisci & Dorn, 2024) offer a useful direction for that larger study.

Reporting implications and limits

A comparison can be made easier to interpret by stating the exact score transformation, invalid-score behavior, within-score and final-aggregate tie conventions, and the feature identities to which priorities attach. Both inner and refit selection decisions matter when a classifier is tuned. Duplicate prevalence should be distinguished from cutoff ambiguity, and prediction, subset stability, and information coverage should retain separate definitions. These details describe the mathematical and statistical procedure being compared, rather than auxiliary file-management conventions.

Selection and scaling were fitted only within each exact training partition. That information boundary follows work on feature-selection bias and nested evaluation (Ambroise & McLachlan, 2002; Varma & Simon, 2006; Vabalas et al., 2019). The order effect diagnosed here is not leakage. Leakage remains a separate threat to machine-learning-based science (Kapoor & Narayanan, 2023), and keeping the boundary correct does not make this result-motivated study confirmatory.

External validity remains limited by four chosen conditions, one synthetic realization, one classifier, fixed k, B=20, a three-point C grid, and one column permutation. They restrict what can be said about other datasets or budgets. Overlapping folds also restrict inferential interpretation (Nadeau & Bengio, 2003; Bengio & Grandvalet, 2004): the absence of naive fold-level intervals does not assert that uncertainty estimation is impossible, but these ten values are not ten independent dataset replications. A larger prospective study could vary selection size, bootstrap budget, score representation, classifier, and data realization under shared evaluation boundaries, then assess the complete procedures rather than seeking a preferred outcome after exposure. Multi-domain and recent application studies (Pes, 2020; Ahmadi & Mahmoodian, 2024; Xiang et al., 2024) illustrate the breadth needed for stronger generalization.

Conclusions

Boundary ambiguity, rather than ties anywhere in a vector, identified where final priorities could change selected membership. The saved frequency bootstrap cutoffs were unique, locating the observed order dependence after vote aggregation. A complete digits trace then showed two inner substitutions changing regularization and predictions despite identical final feature sets. Construction-defined group coverage exposed repeated information omissions despite complete noise exclusion. These results explain conditional differences between established aggregators and support explicit reporting of identities, cutoff conventions, and complete tuning decisions. They do not identify a universally superior selector or a new error-control method.

All four figures use deterministic plotting code assisted by OpenAI Codex/GPT-6.1 Sol (requested model identifier `gpt-6.1-sol`, Codex CLI 0.160.0) and rendered with Matplotlib 3.10.9. They are data visualizations or method diagrams, not generated experimental images.

References

Abeel T, Helleputte T, Van de Peer Y, Dupont P, Saeys Y. 2010. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics* 26:392–398. DOI: 10.1093/bioinformatics/btp630.

Ahmadi M, Mahmoodian H. 2024. Stable feature selection based on probability estimation in gene expression datasets. *Expert Systems with Applications* 248:123372. DOI: 10.1016/j.eswa.2024.123372.

Ambroise C, McLachlan GJ. 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences* 99:6562–6566. DOI: 10.1073/pnas.102102699.

Barbieri MC, Grisci BI, Dorn M. 2024. Analysis and comparison of feature selection methods towards performance and stability. *Expert Systems with Applications* 249:123667. DOI: 10.1016/j.eswa.2024.123667.

Ben Brahim A, Limam M. 2018. Ensemble feature selection for high dimensional data: A new method and a comparative study. *Advances in Data Analysis and Classification* 12:937–952. DOI: 10.1007/s11634-017-0285-y.

Bengio Y, Grandvalet Y. 2004. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research* 5:1089–1105.

Bolón-Canedo V, Alonso-Betanzos A. 2019. Ensembles for feature selection: A review and future trends. *Information Fusion* 52:1–12. DOI: 10.1016/j.inffus.2018.11.008.

Bommert A, Rahnenführer J. 2020. Adjusted measures for feature selection stability for data sets with similar features. In: *Machine learning, optimization, and data science*. Lecture notes in computer science. 203–214. DOI: 10.1007/978-3-030-64583-0_19.

Breiman L. 1996. Bagging predictors. *Machine Learning* 24:123–140. DOI: 10.1007/bf00058655.

Brodersen KH, Ong CS, Stephan KE, Buhmann JM. 2010. The balanced accuracy and its posterior distribution. In: *2010 20th international conference on pattern recognition*. 3121–3124. DOI: 10.1109/icpr.2010.764.

Cawley GC, Talbot NLC. 2010. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research* 11:2079–2107.

Demšar J. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7:1–30.

Ding X, Wang X, Chen S, Wang K. 2026. Multi-level ensemble feature selection for omics data. *Pattern Recognition* 173:112809. DOI: 10.1016/j.patcog.2025.112809.

Guyon I, Elisseeff A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3:1157–1182.

Haury A-C, Gestraud P, Vert J-P. 2011. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PLoS ONE* 6:e28210. DOI: 10.1371/journal.pone.0028210.

Kalousis A, Prados J, Hilario M. 2007. Stability of feature selection algorithms: A study on high-dimensional spaces. *Knowledge and Information Systems* 12:95–116. DOI: 10.1007/s10115-006-0040-8.

Kapoor S, Narayanan A. 2023. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns* 4:100804. DOI: 10.1016/j.patter.2023.100804.

Kolde R, Laur S, Adler P, Vilo J. 2012. Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics* 28:573–580. DOI: 10.1093/bioinformatics/btr709.

Meinshausen N, Bühlmann P. 2010. Stability selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 72:417–473. DOI: 10.1111/j.1467-9868.2010.00740.x.

Melikechi O, Miller JW. 2026. Integrated path stability selection. *Journal of the American Statistical Association* 121:454–464. DOI: 10.1080/01621459.2025.2525589.

Nadeau C, Bengio Y. 2003. Inference for the generalization error. *Machine Learning* 52:239–281. DOI: 10.1023/a:1024068626366.

Nogueira S, Sechidis K, Brown G. 2018. On the stability of feature selection algorithms. *Journal of Machine Learning Research* 18:1–54.

Parlak B. 2023. Ensemble feature selection for single-label text classification: A comprehensive analytical study. *Neural Computing and Applications* 35:19235–19251. DOI: 10.1007/s00521-023-08763-y.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay Édouard. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12:2825–2830.

Pes B. 2020. Ensemble feature selection for high-dimensional data: A stability analysis across multiple domains. *Neural Computing and Applications* 32:5951–5973. DOI: 10.1007/s00521-019-04082-3.

Rakesh DK, Anwit R, Jana PK. 2023. A new ranking-based stability measure for feature selection algorithms. *Soft Computing* 27:5377–5396. DOI: 10.1007/s00500-022-07767-5.

Saeys Y, Abeel T, Van de Peer Y. 2008. Robust feature selection using ensemble feature selection techniques. In: *Machine learning and knowledge discovery in databases*. Lecture notes in computer science. 313–325. DOI: 10.1007/978-3-540-87481-2_21.

Saeys Y, Inza I, Larrañaga P. 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics* 23:2507–2517. DOI: 10.1093/bioinformatics/btm344.

Salman R, Alzaatreh A, Sulieman H. 2022. The stability of different aggregation techniques in ensemble feature selection. *Journal of Big Data* 9:51. DOI: 10.1186/s40537-022-00607-1.

Salman R, Alzaatreh A, Sulieman H. 2026. Within importance score aggregation for wrapper-based feature selection and its stability. *Annals of Data Science* 13:329–353. DOI: 10.1007/s40745-025-00595-y.

Salman R, Alzaatreh A, Sulieman H, Faisal S. 2021. A bootstrap framework for aggregating within and between feature selection methods. *Entropy* 23:200. DOI: 10.3390/e23020200.

scikit-learn developers. SelectKBest API documentation.

scikit-learn developers. Common pitfalls and recommended practices. *Available at* `https://scikit-learn.org/stable/common_pitfalls.html` (accessed October 4, 2026).

scikit-learn developers. Nested versus non-nested cross-validation. *Available at* `https://scikit-learn.org/stable/auto_examples/model_selection/plot_nested_cross_validation_iris.html` (accessed October 4, 2026).

SciPy developers. Rankdata API documentation.

Seijo-Pardo B, Porto-Díaz I, Bolón-Canedo V, Alonso-Betanzos A. 2017. Ensemble feature selection: Homogeneous and heterogeneous approaches. *Knowledge-Based Systems* 118:124–139. DOI: 10.1016/j.knosys.2016.11.017.

Shah RD, Samworth RJ. 2013. Variable selection with error control: Another look at stability selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 75:55–80. DOI: 10.1111/j.1467-9868.2011.01034.x.

Slavkov I, Petković M, Geurts P, Kocev D, Džeroski S. 2020. Error curves for evaluating the quality of feature rankings. *PeerJ Computer Science* 6:e310. DOI: 10.7717/peerj-cs.310.

Vabalas A, Gowen E, Poliakoff E, Casson AJ. 2019. Machine learning algorithm validation with a limited sample size. *PLOS ONE* 14:e0224365. DOI: 10.1371/journal.pone.0224365.

Varma S, Simon R. 2006. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics* 7:91. DOI: 10.1186/1471-2105-7-91.

Xiang F, Zhao Y, Zhang M, Zuo Y, Zou X, Tao F. 2024. Ensemble learning-based stability improvement method for feature selection towards performance prediction. *Journal of Manufacturing Systems* 74:55–67. DOI: 10.1016/j.jmsy.2024.03.001.