WHY ERASED EMBEDDINGS LOOK SAFE: FITTING THE ERASER ON THE ROWS YOU RELEASE

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept erasure is usually evaluated by how far an attacker's accuracy on the erased representation sits above chance. We show that this number depends on two choices that papers rarely report: which family of attackers runs the probe, and whether the eraser was fitted on the same rows that are released. Owners typically fit the eraser on the data they hold, which is also the data they release. We prove that when the embedding has at least as many dimensions as released rows, this fit subtracts each row's own class centroid, and that for any number of rows it leaves a probe trained on one part of the release pointing against the rest. A sweep over seven embeddings, six ratios of rows to dimensions and three erasure methods confirms both consequences. Linear probes fall below chance at all 28 settings, which the standard metric reads as strong erasure. On ReLU embeddings, where more than half of the coordinates are exactly zero, the same fit turns every zero into a class-specific constant, and a depth-4 tree recovers the protected attribute at 0.79 above chance while linear probes sit 0.20 below it. Fitting the eraser on disjoint rows removes both effects. The mechanism also explains why, on 109 evaluation cells with three published erasure methods at matched utility, two attacker families of equal size disagree about whether to release on 9.1% of cases. The disagreement recurs on 73 new cells, and it extends to method rankings: a kernel-based eraser leaks less than LEACE under one attacker family and no less under the other.

1 INTRODUCTION

Releasing embeddings after training is a different privacy problem from private training. The encoder is frozen, the owner releases vectors so that a recipient can train a model on them, and one sensitive attribute should stay unreadable. Linear concept erasure removes the directions a linear probe would use (Ravfogel et al., 2020; Belrose et al., 2023). SPLINCE keeps the linear removal but preserves task covariance with an oblique projection (Holstege et al., 2025), and LEOPARD and Obliviator target nonlinear readout (Saillenfest & Lemberger, 2025; Akbari et al., 2025). After erasure the owner still has to decide whether to ship the release and, when several methods are available, which one to use.

Both decisions rest on measurements: a finite family of attackers, a linear audit on a held-out split, sometimes a geometric score. It is well established that a concept erased against one probe can be recovered by a stronger one (Elazar & Goldberg, 2018; Gonen & Goldberg, 2019; Ravfogel et al., 2022b; Kumar et al., 2022), and we claim no novelty for that. Our contribution is a protocol choice that changes the reported number, an exact identity that explains why, and a measurement of how far release decisions move as a result.

Release decisions depend on the attacker family (Section 3). On 109 cells from 21 encoder–dataset panels we fit LEACE, SPLINCE and LEOPARD at matched task utility and measure each release with two attacker families of equal size, each against its own permutation floor. The release decision changes with the family on 22 of 242 cell–method pairs, more often than seed noise explains. The ordering of these three methods almost never changes, but the ordering of LEACE and Obliviator changes on 12 of the 32 cells where it is resolved.

The cause is how the eraser is fitted (Section 4). Owners, and our harness, fit the eraser on the rows they release. At $n \leq d$ that fit subtracts each row's own class centroid, and for any n it forces

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.

the class-mean difference on a training split of the release to point against the one on the test split. Linear probes then read below chance, which an accuracy-above-majority metric counts as safety. Where coordinates tie, as in ReLU embeddings in which half the entries are zero, the tied entries become class constants that a shallow tree reads off. In a controlled sweep, the two attacker families disagree on 10 of 28 settings when the eraser is fitted in-sample and on 0 of 28 when it is fitted on disjoint rows.

A change of attacker can reverse a study's conclusion (Section 5). On a separate 38-cell table we tested a common-cause claim and a residual-aware release rule, then re-estimated the residual with a tuned attacker from the same family. The claim holds with the untuned attacker and fails with the tuned one, and a matched-n control traces the failure to the smallest panels rather than to the mechanism we had proposed.

Study design. The main analyses were specified in writing before they were run, with dated amendments; Appendix E lists them. Two parts were not: the analysis of Section 3 was specified after an exploratory look at the same table (its replication on new cells was specified before those cells were measured), and the framing of Section 5 was adopted after its gate failed. We treat those parts as measurements rather than confirmatory tests.

Contributions.

1. *A measurement.* On a designed table of published erasure methods, release decisions depend on which of two equal-size, floor-corrected attacker families measures them. The effect replicates, more weakly, on new cells, and it reaches method rankings once a kernel-based eraser is compared with a linear one.
2. *A mechanism.* Two identities and a sweep show that fitting the eraser on the released rows drives linear readings below chance and, on embeddings with tied coordinates, hands the attribute to a shallow tree. Fitting on disjoint rows removes the family disagreement.
3. *A reversal* of a tested conclusion under a change of attacker, with a matched-n control and its null distribution.
4. *Reporting practice* that follows from these results (Section 7).

2 Related work

Removing protected attributes. Early work learned representations that hide a protected attribute by training the encoder against an adversary (Zemel et al., 2013; Edwards & Storkey, 2016; Xie et al., 2017; Madras et al., 2018). Elazar & Goldberg (2018) and Barrett et al. (2019) showed that a classifier trained after the fact still recovers the attribute, so the adversary used during training understates leakage. Projection methods act on a frozen representation: hard debiasing removes a bias direction (Bolukbasi et al., 2016), which leaves the attribute recoverable from the remaining geometry (Gonen & Goldberg, 2019); INLP iterates linear projections (Ravfogel et al., 2020); RLACE finds the subspace through a minimax game (Ravfogel et al., 2022a); and LEACE gives the closed-form projection that prevents every linear predictor from recovering the concept at least cost to the representation (Belrose et al., 2023). Spectral, kernel and rate-distortion methods extend removal beyond the linear case (Shao et al., 2023; Iskander et al., 2023; Chowdhury & Chaturvedi, 2022; Chowdhury et al., 2023), as do the three recent methods we evaluate. SPLINCE zeroes covariance with the attribute while preserving covariance with the task (Holstege et al., 2025); LEOPARD matches class-conditional densities under an MMD objective with an adjustable-rank projection (Saillenfest & Lemberger, 2025); and Obliviator erases in a reproducing-kernel Hilbert space and reports that earlier nonlinear methods remain recoverable by stronger adversaries (Akbari et al., 2025).

What erasure guarantees. Linear guardedness bounds linear predictors only. A concept erased against linear probes can be recovered in kernel space (Ravfogel et al., 2022b), and log-linear guardedness does not carry over to multi-class or nonlinear readouts (Ravfogel et al., 2023). Kumar et al. (2022) show that erasure driven by probing classifiers is unreliable because the probe determines what is removed and what is detected. Section 4 describes a failure of a different kind, one of

estimation rather than of adversary class: when the eraser is fitted on the rows it releases, the linear guarantee holds exactly on those rows while the fit writes the label into them. Fitting on training data and evaluating on held-out data is standard practice, and Naowarat et al. (2026) report that LEACE does not generalise to unseen rows; we did not find a prior account of what in-sample fitting does to the released rows themselves.

Probing and leakage measurement. Probe accuracy depends on the probe. Control tasks (Hewitt & Liang, 2019), description-length probing (Voita & Titov, 2020) and information-theoretic analyses (Pimentel et al., 2020) all separate what a representation encodes from what a given probe can extract, and Belinkov (2022) surveys these issues. Leakage in fairness work is likewise defined as the accuracy of an attacker classifier (Wang et al., 2019). We read every attacker against a permutation floor (Ojala & Garriga, 2010) and compare two attacker families of equal size, so that neither wins by taking a maximum over more models.

Privacy of released embeddings. General-purpose embeddings leak sensitive attributes and can often be inverted (Song & Raghunathan, 2020; Pan et al., 2020; Morris et al., 2023), which is why owners erase before release. Our concern is how that erasure is then evaluated.

Evaluation practice. Reported results depend on tuning budgets, seeds and data splits (Dodge et al., 2019; Bouthillier et al., 2021), and evaluation choices can inflate claims (Lipton & Steinhardt, 2019). In watermarking, attacks adapted to each scheme remove marks that fixed attacks leave intact (Diaa et al., 2025). Specifying analyses before running them has been proposed for predictive modelling (Hofman et al., 2023); we follow that practice and report where we could not.

Implementations. LEACE and SPLINCE are implemented to their stated properties (LEACE's covariance condition holds to 1.4×10^{-7}; SPLINCE's two Theorem 1 constraints hold to machine precision). LEOPARD is our reimplementation, which we call LEOPARD (paper-spec); it reproduces the published GloVe Table 1a result 1.5 points weaker (53.3 against 51.8 ± 0.9, i.e. weaker erasure), so every LEOPARD number here describes that reimplementation. Obliviator (paper-spec) runs the authors' released code with its iteration count as the utility dial; we checked it only with a synthetic property test and did not reproduce its published tables.

3 Release decisions under a change of instrument

Table and methods. The designed table has 109 cells (encoder, dataset, task, sensitive attribute) from 21 panels: CLIP, DINOv2, SigLIP and ResNet-50 on FairFace, UTKFace and CelebA; text encoders on 20 Newsgroups and Bias in Bios; tabular ACS; and smaller audio, video, image and synthetic panels. Cells were allocated by a fixed rule, before any method was fitted, across five strata of *affinity* (the mean squared principal cosine between the task and sensitive subspaces, which sets the cost of erasure) and two of *nesting* (how far the task label is determined by the sensitive one). Each method is fitted at the owner's matched task cost (tolerance 0.02) with 30 seeds, and leakage is read only after the match. A cell–method pair enters if at least 20 seeds matched; 242 of 327 do (SPLINCE loses 37 cells and LEOPARD 48, where utility could not be matched).

Instruments. Family R is logistic regression ($C \in \{0.1, 1, 10\}$) and a one-hidden-layer MLP (widths $32, 64, 128$); family T is gradient-boosted trees (depth $2, 4, 8$) and k-NN ($k \in \{3, 10, 30\}$). Both have six members, and each is read against its own permutation floor at the same family size, test size and class balance. A release is approved if leakage above the floor is below $t{=}0.05$. Table 1 collects the main estimates of this section with their intervals.

Decisions. R and T disagree on 22 of 242 cell–method pairs (9.1%; Figure 1a). Some disagreement near the threshold is seed noise, so we split each pair's seeds in half 200 times and compare disagreement across families on different halves (0.096) with disagreement within one family on different halves (0.020). The difference, which we call the excess, is 0.076 and stays positive at $t{=}0.02$ and $t{=}0.10$ (Figure 1b) and when any single panel is dropped. It is largest for LEACE (0.119), smaller for SPLINCE (0.042) and LEOPARD (0.039), and larger on low-nesting cells (0.240) than on high-nesting ones (0.067). 8 of the 22 flips are ResNet-50 cells on FairFace and UTKFace, where

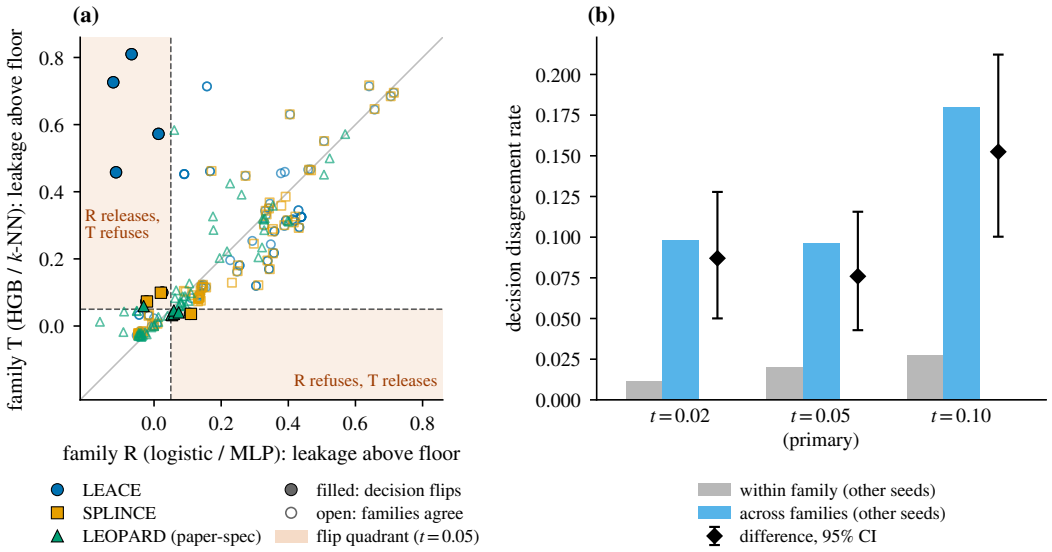

Figure 1: Release decisions under two attacker families of equal size on the designed table. (a) One point per cell–method pair (242); axes are leakage above each family's permutation floor. Shaded quadrants are decision flips at t=0.05 (filled markers). (b) Disagreement between families on disjoint seed halves (blue) against disagreement within one family on disjoint halves (grey), and their difference with a cell-bootstrap 95% interval, at three thresholds; t=0.05 is the primary one.

Table 1: Main estimates on the designed tables. Excess is cross-family minus within-family disagreement on disjoint seed halves; intervals are 95% cell bootstraps. "New cells" are the 73 constructed-label cells added later; "all" is the extended 160-cell table.

Quantity	Table	Count	Estimate [95% CI]
Decision flips, excess	original	22/242	0.076 [0.043, 0.116]
without ResNet-50	original	14/229	0.045 [0.016, 0.082]
Decision flips, excess	new cells	4/155	0.023 [0.004, 0.048]
Decision flips, excess	all	20/350	0.046 [0.025, 0.070]
Order reversals, three projections	original	3/183	—
Order reversals, LEACE–Obliviator	all	12/32	0.227 [0.116, 0.347]
Spearman of R and T leakage	all	159 cells	0.465 [0.286, 0.631]
Linear audit passes that leak	original	156/239	65% [57, 74]%

R reads below its floor and T up to 0.81 above it; all of these are R approving what T refuses, and Section 4 explains why. Without the two ResNet-50 panels, a check we added after scoring, 14 flips remain and the excess is 0.045; the remaining flips split evenly between directions, so neither family is a conservative default.

Rankings. Resolved orderings of two methods reverse between R and T on 3 of 183 cell–pair comparisons (Table 3 in Appendix O); more often an ordering resolved under one family is a tie under the other. The one large difference on this table, LEACE leaking more than LEOPARD, has about the same size under both families: a median gap of 0.067 above floor under R and 0.065 under T. Among these three methods, the comparison survives a change of family while the release decision does not.

The linear audit. A linear probe on a held-out split certifies 239 of the 242 releases (reading at most 0.02 above chance). For LEACE and SPLINCE this is guaranteed by construction, and 65% of certified releases leak more than 0.10 above floor to R or T (63% without the ResNet-50 panels). Passing the audit confirms the linear guarantee but says little about the release.

New cells. We later extended the table to 160 cells: the 87 natural-label cells above, measured again, and 73 cells whose labels are products of binary attributes and which no earlier analysis had used (Appendix B). On the new cells the effect is weaker: 4 of 155 pairs flip, the excess is 0.023, it depends on the threshold (0.000 at t=0.02, 0.118 at t=0.10), and its lower bound reaches zero when either of two FairFace panels is dropped. All 4 flips run in the direction Section 4 predicts, orderings reverse on 1 of 57 resolved comparisons, and 94% of the releases the linear audit certifies leak. Over all 160 cells the excess is 0.046 and survives dropping any panel.

A kernel-based eraser. On the extended table we also fitted Obliviator at matched utility (Appendix C). It too is fitted on the rows it releases, but it is nonlinear, so the identities of Section 4 do not cover it. It reaches the utility target on only 38% of rows, leaving 39 usable cells, and on those cells release decisions barely move for any method (2 Obliviator flips; LEACE's excess on the same cells is 0.051). Rankings do move. LEACE and Obliviator swap places between R and T on 12 of 32 resolved cells, in 10 of them with Obliviator ahead under R and behind under T; SPLINCE and Obliviator swap on 10 of 20, and LEOPARD and Obliviator on none of 19. The median gap between Obliviator and LEACE is -0.040 under R and -0.009 under T, so whether the nonlinear method erases more than the linear one depends on the attacker family. Of the 36 Obliviator releases the linear audit certifies, 29 leak.

Does one family predict the other? An owner could use family R as a proxy for T if the two ranked releases consistently. We fixed a bar for this in advance: a Spearman correlation across cells of at least 0.55, with a lower bound of at least 0.40 (Appendix D). On 159 cells the correlation is 0.465, 0.509 on natural labels alone and 0.451 after controlling for sample size, so the bar is not met. Ranking candidate releases by what R reads recovers T's ranking only in part.

4 WHY THE FAMILIES DISAGREE: FITTING THE ERASER ON THE ROWS IT RELEASES

In Section 3, as in an owner's workflow, every eraser is fitted on the rows that are then released and attacked. This section shows that this choice accounts for most of the disagreement measured there, that it has an exact algebraic signature, and that its effect on measured leakage runs in opposite directions on different encoders. The sweep covers 7 cached embeddings, six ratios n/d from $1/4$ to 8, three erasers, three fitting protocols and three seeds, and tests ten statements (H1–H10) fixed before it ran; four hold, and all ten are reported in Appendix G.

The scenario. An owner holds n rows with labels for the attribute to be protected, fits an eraser with those labels, and releases the same rows; the recipient trains on what arrives. This is the setting the erasure literature targets and the one our designed table implements. Erasers are usually *evaluated* differently: the eraser is fitted on a training split and measured on held-out rows, which we call disjoint fitting. An attacker holds the attribute for part of the release and infers it for the rest, so both the eraser and the attacker see the released rows.

Two identities. Let X be the n released rows, μ their mean and μ_c their class means.

(i) At $n \le d+1$, in-sample LEACE subtracts each row's own class centroid. If the centred rows have rank $n-1$, the fitted map sends each released row to $x_i - (\mu_{y_i} - \mu)$ (proof in Appendix A). On the 36 configurations with $n \le d/2$ this holds to a relative 8.5×10^{-14}. At $n = d$ the covariance is ill-conditioned and the deviation reaches 0.029, so H1, stated at a tolerance of 10^{-6} for all rank-$(n-1)$ configurations, fails as stated.

(ii) In-sample erasure makes any train/test split of the released rows anti-correlated. If the class means are equal over all n rows, then the per-class centred sums of any two complementary subsets cancel, so the class-mean difference on an attacker's training split is antiparallel to the one on its test split. The cancellation of the sums is exact; the class-*mean* differences are exactly antiparallel only when both parts hold the same per-class proportions, which a stratified split gives up to integer rounding. On the two binary caches the class sums cancel to 1×10^{-4} and the cosine differs from -1 by at most 0.0013. H10, stated at 10^{-6} on the cosine, fails because of that rounding and the conditioning that also limits H1.

Consequence 1: linear readings fall below chance. A probe fitted on part of a release and tested on the rest points the wrong way. On all 7 caches and at every ratio, 28 points in all, in-sample logistic leakage is negative, from -0.116 to -0.453 (H7 holds), while the same release fitted on disjoint rows reads 0.016 to 0.171 (Figure 2b). The members of family R are not affected equally: logistic regression is negative at all 28 of 28 points (median -0.207) and the MLP at 15 (median -0.043), against 12 for the tree and 7 for k-NN. A metric that scores accuracy above the majority rate therefore reports these releases as better than erased. The negative reading cannot be exploited: after learning a label remap on held-back training rows, in-sample leakage returns to -0.006 (median) against 0.059 for disjoint fitting, so H8, which predicted that a remapping attacker would beat disjoint fitting, fails. The anti-correlation is a constraint on the release as a whole rather than a stable direction: a remap learned on an internal sub-split is fitted against that sub-split's complement, not the one the test split sees. In-sample fitting does remove linear leakage from the released rows; the number an auditor reads for it, a median -0.156, is simply not that leakage.

Consequence 2: where coordinates tie, the class becomes a constant. By identity (i), every row with $x_{ij} = v$ leaves the eraser with the value $v - (\mu_{c,j} - \mu_j)$, which identifies its class. ReLU embeddings have many ties: 52% and 61% of the entries of the two ResNet-50 caches are exactly zero, against none of the dense ones. At $n/d = 1/4$ a depth-4 tree recovers race from the in-sample release at 0.786 and 0.517 above majority, against 0.104 and 0.102 when the same eraser is fitted on disjoint rows (Figure 2a), and both sparse caches exceed every dense one (H4 holds). The leak fades beyond $n/d = 1$, where identity (i) no longer holds and the class code is only approximate. Comparing erasers confirms the mechanism. SAL (Shao et al., 2023), an unwhitened projection on the class-mean difference that cannot interpolate the released rows, gives gaps of 0.032 and -0.019, while SPLINCE, which whitens as LEACE does, gives 0.660 and 0.429.

Ties, not ReLU. The sparse caches differ from the dense ones in their encoder as well as in their ties, so we created ties directly: we quantised a dense embedding to b bits per coordinate before erasing, leaving the encoder unchanged. On FairFace CLIP and DINOv2 the tree gap stays negative down to 6 bits (tie density 0.070), turns positive at 4 bits (tie density 0.214) and reaches 0.716 and 0.766 at 2 bits, where task accuracy is within 0.05 of the unquantised release. The hazard therefore follows ties rather than the activation function. Eight-bit quantisation, the usual deployment choice, leaves tie density at 0.026–0.030 and shows no leak. Two of the five statements for this experiment fail, both about the shape we expected rather than the effect: the rise is a threshold rather than a monotone trend (Spearman 0.706 on CLIP and 0.816 on DINOv2 against a bar of 0.80), and the SAL control was stated as $|\text{gap}| < 0.02$, which in-sample suppression breaks for any eraser. By sign, SAL behaves as a control should: negative at every level down to 4 bits (at most -0.101) and positive only at 2 bits (0.293), so whitening amplifies the leak, while extreme ties eventually produce it on their own (Appendix F).

The primary hypothesis failed. H2 asked whether in-sample fitting raises the maximum attacker reading at $n \leq d$ across caches. The median gap is -0.056 and the hypothesis fails. On the dense caches the maximum reading is *lower* under in-sample fitting at 17 of 26 points and higher by at most 0.207 at the other 9; the two sparse caches raise it by 0.405 and 0.660. The transition hypothesis H3 fails for the same reason, and two-fold cross-fitting (H5) does not help: it leaves the maximum reading nan below the disjoint-fit reading at no utility cost (nan). Among the protocols we tested, only fitting on rows disjoint from the release gives the behaviour a reader would assume. The effect is not limited to small samples: at $n/d = 16$ on Bios MiniLM ($n = 6144$ rows), the in-sample logistic reading stays below the majority rate on 3 of 3 seeds (-0.116, against 0.016 for disjoint fitting), while the maximum-attacker gap closes to 0.005. The measurement artefact weakens with n but persists; the exploitable leak does not.

Connection to Section 3. Reducing each family to its strongest member on this sweep, the two families disagree about release on 10 of 28 points when the eraser is fitted in-sample and on 0 of 28 when it is fitted on disjoint rows (H9 holds). Every in-sample disagreement has the same sign: the logistic/MLP family approves and the tree/k-NN family refuses (Figure 2c). Most of the instrument dependence in Section 3 is thus the signature of fitting on the released rows, which is what the designed table, and an owner erasing the data at hand, actually does.

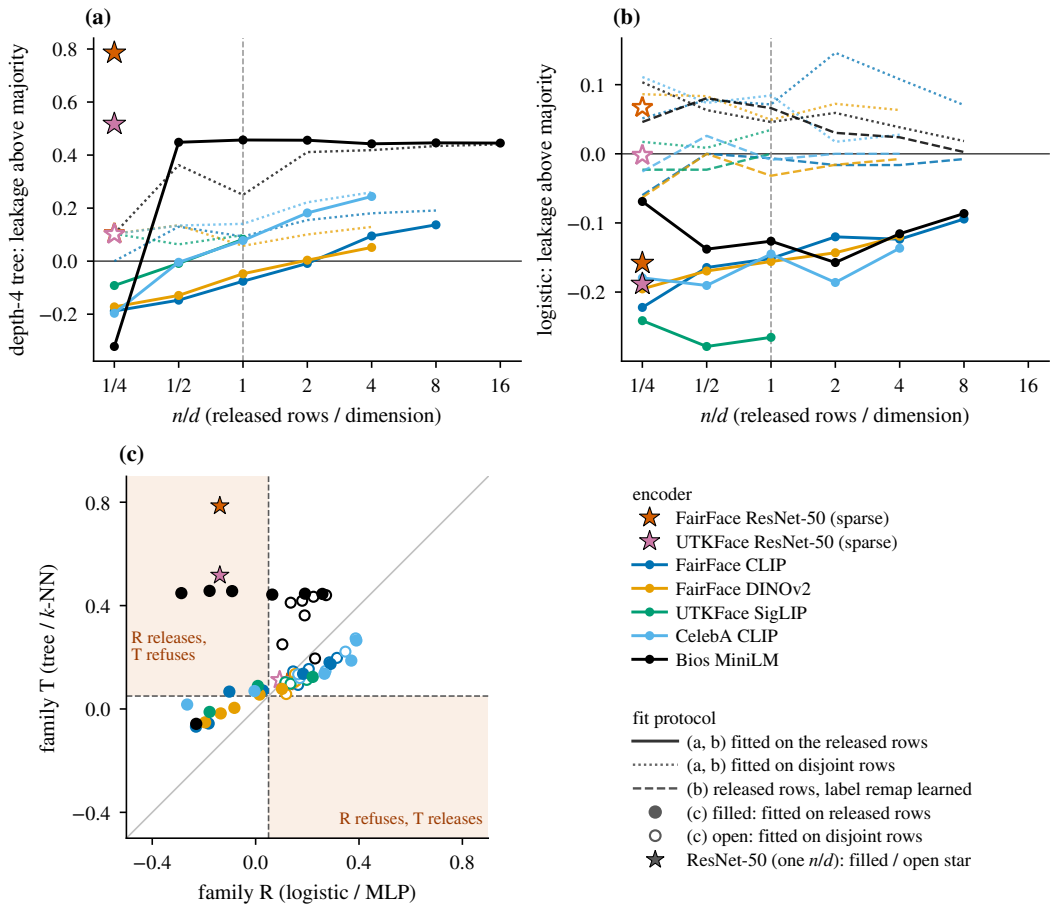

Figure 2: Fitting the eraser on the released rows against fitting it on disjoint rows (LEACE, 7 caches, 3 seeds). (a) A depth-4 tree: on the sparse ResNet-50 embeddings (stars; 52% and 61% of entries are exact zeros) in-sample fitting hands over the attribute, while on dense embeddings it suppresses the reading. (b) Logistic regression: in-sample readings are below the majority rate everywhere (solid) and return to about zero once a label remap is learned (dashed), while disjoint fitting reads positive (dotted). (c) The same releases seen by the two attacker families of Section 3: in-sample points (filled) fall in the disagreement quadrant, disjoint-fit points (open) do not.

5 A TESTED CONCLUSION REVERSED BY A CHANGE OF ATTACKER

The designed table shows that decisions move. This section shows what that looks like inside a single study: a conclusion we specified in advance, a change of instrument that reverses it, and a control that attributes the reversal to the evaluation table rather than to the mechanism we had proposed.

Setup. 44 cells at 30 seeds, with 38 remaining once saturated cells (affinity ≥ 0.99) are dropped and no panel is allowed more than 25% of the table. For each cell the residual R is the maximum leakage above chance over the attacker suite after erasure, with the estimator fixed before any test ran. Gate F-A states a common-cause claim: the residual should organise both the error of a geometric cost predictor and the error of a linear audit (C1: Spearman of R with the geometry error, bar 0.60). *Rung 1* estimates R with the untuned two-attacker suite that defined it; *rung 2* searches hyperparameters within the same family (14 configurations). A rule fixed before rung 2 existed triggers a re-score if any cell moves by more than 0.02.

The gate reverses. Rung 2 moved 26 of 44 cells by more than 0.02 (max $|\Delta R|$=0.374). Gate F-A passes at rung 1 (C1 0.663) and fails at rung 2 (C1 0.580, and C4 fails; Table 2). C1 over 38 cells

Table 2: Gate F-A at two residual rungs (n=38). C4 passes when the partial correlation is below half the raw one, which is 0.314 at C3 = 0.627.

Criterion	Rung 1 (untuned)	Rung 2 (tuned)
C1 (Spearman R, e_{geom}), bar 0.60	0.663	0.580
C4 (partial $e_{\text{geom}}, e_{\text{audit}} \mid R$), bar < 0.314	0.216	0.393
`all_pass`	true	false

has a standard error near 0.11, so we do not claim that the association itself fell; the point is that the verdict of the procedure depends on which rung measured R. Leave-one-panel-out C1 spans 0.485–0.665, and dropping either of the two video panels alone clears the bar (Appendix H).

Two candidate mechanisms. The failure concentrates on two video panels, which are both the only concatenated packing and the smallest panels (n=128). *Composition* (Gate G-C) fails its own test: at 30 seeds, the median ΔR over 11 fresh concatenations (0.085) is 1.72× that of 22 native blocks (0.050), against a bar of 3× (p=0.26), and the two controls pass 1 of 4 and 2 of 5; an earlier 3-seed run gave the same verdict (Appendix J). *Sample size* (Gate G-D) subsamples rows on the two panels that have enough of them and re-estimates both rungs at 30 seeds. Scored against chance, it passed. The rungs, however, take maxima over families of different sizes (2 against 14), so ΔR is positive in expectation even with no leakage, and more so at small n; a null could pass the test. We measured each family's floor by permutation and scored again relative to it (Gate G-D2, Figure 6 in Appendix H). The excess still falls monotonically with n ($\rho = -0.800$ and $\rho = -1.000$) and reaches 0.158 and 0.193 at n=128. A control cell with 0.825 of headroom (the room between a cell's untuned residual and the ceiling its chance level allows) does not move. At matched family size the excess survives on the image panel (0.209) and falls to 0.095 on the text panel, so tuning and family size are partly separable.

The release rule. A residual-aware release rule (Gate F-B) cuts leave-one-panel-out false releases from 0.545 (geometry alone) to 0.182, and a paired bootstrap excludes zero. It stays above the deployment bar of 0.10 at every label budget up to 4000 rows (asymptote 0.182). One rule-by-rung configuration reaches 0.091, but it is the minimum of six rates, relies on two choices made after the fact and has an interval reaching 0.182, so we do not count it as a pass (Appendix L). Adopting the tuned rung where it helps the rule would also mean adopting it where it fails the gate.

6 Discussion and limitations

Our main results concern the protocol an owner follows rather than the erasure methods themselves. The designed table fits every eraser on the rows it releases, and the sweep suggests that the family disagreement would largely disappear with disjoint fitting (H9). A designed table with a disjoint-fit arm would test this directly, and we see it as the most useful next measurement.

The identities explain the below-chance linear readings and the leak through tied coordinates, but not why in-sample fitting usually lowers the tree and k-NN readings on dense embeddings; that sign remains open. The sweep covers LEACE, SPLINCE and a mean-difference projection, so whether INLP, RLACE, LEOPARD or Obliviator show the tied-coordinate leak is untested. Obliviator reaches the matched utility on only 38% of rows, and its 39 usable cells are ones where no method's decision moves much, so its rankings are informative while its release decisions are not.

The gate study is small. With 44 cells a single cell moves a false-release rate by 2.27 points, which is too coarse to resolve a deployment bar of 0.10, and several panels lack the rows to test the common-cause claim on a uniformly large table. With the panels we have, we cannot separate sample size from headroom as the driver of the tuning gains (Appendix M), and one text panel (`20ng-nested-minilm`) moves 0.324 at full sample size for reasons we cannot identify. More broadly, attacker families that differ in model class, not only in hyperparameters, seem to us the right unit for evaluating erasure, and evaluation tables that include both fitting protocols by design would make results comparable across papers.

7 REPORTING ERASURE EVALUATIONS

Each recommendation below would have changed a number in this paper.

1. **State whether the eraser was fitted on the evaluated rows, and report both protocols.** Here the choice moves the measured leakage by -0.264 to 0.660 depending on the encoder, and it is the main source of disagreement between our two attacker families.
2. **Do not read a below-chance probe as erasure.** Under in-sample fitting it is the expected reading at every n/d we tried; the underlying leakage is about zero, not negative.
3. **Check ties and n/d before releasing a linearly erased embedding.** Tied coordinates, $n \leq d$ and a whitening eraser together hand the attribute to a shallow tree. Ties come from ReLU sparsity and from coarse quantisation alike; 4 bits per coordinate is enough, 8 bits is not.
4. **Measure with at least two attacker families of equal size**, each against its own permutation floor, and report both the decision and any method ranking under each. One family does not predict the other well enough to replace it.
5. **Fix the attacker tuning ladder in advance and report every rung.** A tuned search within the same family moved 26 of 44 cells and reversed a conclusion.
6. **Before attributing an effect to a mechanism, run a matched-n control together with its own null.** Ours first passed inside its null.

None of these requires a new method, and the first costs only a data split.

8 CONCLUSION

Whether an erased embedding looks safe to release depends on which attacker family measures it. On our tables, most of that dependence comes from a choice that papers do not report: whether the eraser was fitted on the rows being released. In-sample fitting leaves a recognisable signature, below-chance linear probes and class constants wherever coordinates tie, and it moves the measured leakage by -0.264 to 0.660 depending on the encoder. The dependence extends to method rankings when a kernel-based eraser is compared with a linear one, and one attacker family predicts the other too weakly to stand in for it. Reporting the fitting protocol and measuring with more than one attacker family would make erasure results considerably easier to interpret.

AI USE STATEMENT

Generative AI tools were used to help write and edit the manuscript, to find related work, and to write and debug analysis code. AI assistance also proposed the two identities in Section 4 and drafted the proof in Appendix A; the author checked the proof and verified both identities numerically. All numbers in the paper are produced by scripts from recorded results, and the author takes full responsibility for the content.

ETHICS STATEMENT

This work studies privacy measurement on public research datasets and cached embeddings (20 Newsgroups, Bias in Bios, ACS, and cached embeddings of FairFace, UTKFace and CelebA images). No human subjects were recruited, and we release no new personally identifiable data. The risk we study is attribute recovery from embeddings that owners already choose to share; the paper argues for more careful evaluation of that risk rather than presenting a new attack on individuals. Demographic attributes (race, gender, age) are used only as the labels an owner would try to protect, and we report aggregate leakage rates, not per-person predictions.

REPRODUCIBILITY STATEMENT

The supplementary material contains the code, the analysis specifications with their dated amendments, the recorded results and the scripts that generate every number, table and figure in the paper

from them.[1] Section 3 and Appendix E describe the attacker families, thresholds, seeds and bootstrap settings; the proof of identity (i) is in Appendix A.

REFERENCES

Reza Akbari, Mohammad Afshari, and Vishnu Naresh Boddeti. Obliviator reveals the cost of nonlinear guardedness in concept erasure. In *NeurIPS*, 2025. arXiv:2603.07529.

Maria Barrett, Yova Kementchedjhieva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. Adversarial removal of demographic attributes revisited. In *Proceedings of EMNLP-IJCNLP 2019*, pp. 6329–6334, 2019. doi: 10.18653/v1/D19-1662.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Rafał Cywiński, Edgar Dobriban, and Jacob Steinhardt. LEACE: Perfect linear concept erasure in closed form. In *NeurIPS*, 2023.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Dmitriy Serdyuk, Tal Arbel, Chris Pal, Gaël Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, volume 3, 2021.

Somnath Basu Roy Chowdhury and Snigdha Chaturvedi. Learning fair representations via rate-distortion maximization. *Transactions of the Association for Computational Linguistics*, 10: 1159–1174, 2022. doi: 10.1162/tacl_a_00512.

Somnath Basu Roy Chowdhury, Nicholas Monath, Kumar Avinava Dubey, Amr Ahmed, and Snigdha Chaturvedi. Robust concept erasure via kernelized rate-distortion maximization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 43284–43306, 2023.

Abdulrahman Diaa, Toluwani Aremu, and Nils Lukas. Optimizing adaptive attacks against watermarks for language models. In *ICML*, volume 267 of *PMLR*, pp. 13546–13569, 2025.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. Show your work: Improved reporting of experimental results. In *Proceedings of EMNLP-IJCNLP 2019*, pp. 2185–2194, 2019. doi: 10.18653/v1/D19-1224.

Harrison Edwards and Amos Storkey. Censoring representations with an adversary. In *International Conference on Learning Representations (ICLR)*, 2016.

Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. In *Proceedings of EMNLP 2018*, pp. 11–21, 2018. doi: 10.18653/v1/D18-1002.

Hila Gonen and Yoav Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *NAACL*, 2019.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of EMNLP-IJCNLP 2019*, pp. 2733–2743, 2019. doi: 10.18653/v1/D19-1275.

Jake M. Hofman, Angelos Chatzimparmpas, Amit Sharma, Duncan J. Watts, and Jessica Hullman. Pre-registration for predictive modeling. *arXiv preprint arXiv:2311.18807*, 2023.

Felix Holstege, Shauli Ravfogel, and Bram Wouters. Preserving task-relevant information under linear concept removal. In *NeurIPS*, 2025. arXiv:2506.10703.

[1] Anonymized archive: https://drive.google.com/file/d/1gQwdnkgniqKuPNxNT9upLwQtdAzPdQ5Z.

Shadi Iskander, Kira Radinsky, and Yonatan Belinkov. Shielded representations: Protecting sensitive attributes through iterative gradient-based projection. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5961–5977, 2023. doi: 10.18653/v1/2023.findings-acl.369.

Abhinav Kumar, Chenhao Tan, and Amit Sharma. Probing classifiers are unreliable for concept removal and detection. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17994–18008, 2022.

Zachary C. Lipton and Jacob Steinhardt. Troubling trends in machine learning scholarship. *Queue*, 17(1):45–77, 2019. doi: 10.1145/3317287.3328534.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *PMLR*, pp. 3384–3393, 2018.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. Text embeddings reveal (almost) as much as text. In *Proceedings of EMNLP 2023*, pp. 12448–12460, 2023. doi: 10.18653/v1/2023.emnlp-main.765.

Burin Naowarat, Hao Tang, and Sharon Goldwater. A framework for analyzing concept representations in neural models. In *CoNLL*, 2026.

Markus Ojala and Gemma C. Garriga. Permutation tests for studying classifier performance. *Journal of Machine Learning Research*, 11:1833–1863, 2010.

Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *IEEE Symposium on Security and Privacy*, pp. 1314–1331, 2020. doi: 10.1109/SP40000.2020.00095.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In *Proceedings of ACL 2020*, pp. 4609–4622, 2020. doi: 10.18653/v1/2020.acl-main.420.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *ACL*, 2020.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan Cotterell. Linear adversarial concept erasure. In *ICML*, volume 162 of *PMLR*, pp. 18400–18421, 2022a.

Shauli Ravfogel, Francisco Vargas, Yoav Goldberg, and Ryan Cotterell. Adversarial concept erasure in kernel space. In *EMNLP*, 2022b.

Shauli Ravfogel, Yoav Goldberg, and Ryan Cotterell. Log-linear guardedness and its implications. In *ACL*, 2023.

Antoine Saillenfest and Patrick Lemberger. Nonlinear concept erasure: a density matching approach. *arXiv preprint arXiv:2507.12341*, 2025.

Shun Shao, Yftah Ziser, and Shay B. Cohen. Gold doesn't always glitter: Spectral removal of linear and nonlinear guarded attribute information. In *Proceedings of EACL 2023*, pp. 1611–1622, 2023. doi: 10.18653/v1/2023.eacl-main.118.

Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 377–390, 2020. doi: 10.1145/3372297.3417270.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In *Proceedings of EMNLP 2020*, pp. 183–196, 2020. doi: 10.18653/v1/2020.emnlp-main.14.

Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5309–5318, 2019. doi: 10.1109/ICCV.2019.00541.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *PMLR*, pp. 325–333, 2013.

A PROOF OF THE CENTROID IDENTITY

Let $X \in \mathbb{R}^{n\times d}$ be the released rows with mean μ, centred matrix X_c of rank $n-1$, and class indicators Z with centred form Z_c. LEACE fits $\hat{\Sigma} = X_c^\top X_c/n$, its whitening $W = \hat{\Sigma}^{+1/2}$ on the support, and removes the column space of $W\hat{\Sigma}_{xz}$ in whitened coordinates, returning $x \mapsto x - W^+PW(x-\mu)$, where P is that projection.

Write $\Omega = WX_c^\top$, whose columns are the whitened centred rows. Then $\Omega\Omega^\top/n$ is the projector onto the support of $\hat{\Sigma}$, and because $\mathrm{rank}(X_c) = n-1$ and $X_c^\top \mathbf{1} = 0$, $\Omega^\top\Omega/n$ is the centering matrix $C = I - \mathbf{1}\mathbf{1}^\top/n$. The whitened cross-covariance is $\Omega Z_c/n$, so $P = \Omega Z_c(Z_c^\top\Omega^\top\Omega Z_c)^+ Z_c^\top\Omega^\top$. For the i-th released row, using $CZ_c = Z_c$ and $Ce_i = e_i - \mathbf{1}/n$,

$$P\Omega e_i = \Omega Z_c(Z_c^\top Z_c)^+ Z_c^\top e_i = \Omega(He_i - \mathbf{1}/n),$$

where H averages within a class. The right-hand side is the whitened class centroid $W(\mu_{y_i} - \mu)$. Applying W^+, and noting that $\mu_{y_i} - \mu$ lies in the support, gives $x_i \mapsto x_i - (\mu_{y_i} - \mu)$. □

The argument needs $\mathrm{rank}(X_c) = n-1$, which holds for points in general position whenever $n \le d+1$, *including* $n = d$. So the identity is exact in arithmetic at $n = d$, and the deviation we measure there (0.029, against 8.5×10^{-14} at $n \le d/2$) is not a rank failure: it is the ill-conditioning of $\hat{\Sigma}^{+1/2}$ when the smallest retained eigenvalue approaches the tolerance. Past $n = d+1$ the rank condition genuinely fails, the map can no longer realise a per-row edit, and the identity stops holding.

The second identity needs no rank condition. Because in-sample erasure leaves the released rows with equal class means, the centred rows of each class c sum to zero, so for any split into A and its complement B,

$$\sum_{i\in A,\, y_i=c} (x_i - \mu) \;=\; -\sum_{i\in B,\, y_i=c} (x_i - \mu).$$

The class *sums* are therefore exact negatives for every split. Dividing by the per-class counts, the class-mean difference of A is $-(n_1^B/n_1^A)\,\bar{\delta}_1^B + (n_0^B/n_0^A)\,\bar{\delta}_0^B$ in the binary case, so it is antiparallel to B's exactly when $n_1^B/n_1^A = n_0^B/n_0^A$: the two parts must hold the same per-class proportions. A stratified split satisfies this up to integer rounding, which is why the measured cosine sits within 0.0013 of -1 rather than at it. Either way, a probe fitted on one part is aimed against the other.

B REPLICATION ON THE EXTENDED DESIGNED TABLE (X8)

Declared in `research/PREREGISTRATION_x8_replication.md` before any v2 decision was computed; scored by `scripts/score_x8_replication.py`, which calls the unchanged scorer of Section 3. Excess is cross-family minus within-family disagreement at t=0.05 with a cell-bootstrap 95% interval; LOPO is the minimum leave-one-panel-out point and the number of dropped panels that take the lower bound to zero.

Subset	cells	pairs	flips (R/T)	excess [95% CI]	LOPO min (fails)	orderings reversed
73 unseen (primary)	73	155	4 (4/0)	0.023 [0.004, 0.048]	0.010 (2)	1/57
87 re-measured	87	195	16 (12/4)	0.063 [0.032, 0.105]	0.044 (0)	3/56
all 160	160	350	20 (16/4)	0.046 [0.025, 0.070]	0.030 (0)	4/113

Declared readings: the primary (lower bound above zero on the unseen cells) pass; leave-one-panel-out robustness fail (the lower bound reaches zero without fairface dinov2, fairface resnet50); the predicted direction of the flips pass; method orderings reversing on at most 5% of resolved comparisons pass. The largest panel supplies 47% of the unseen cells (CelebA), because constructed labels are easiest to build from its forty binary attributes.

C A KERNEL-SPACE ERASER ON THE EXTENDED TABLE (X9)

Declared in `research/PREREGISTRATION_obliviator_x9.md` before any Obliviator reading was aggregated, with the X1, X2 and X4 estimators unchanged and the arm joined to the three-arm

rows by cell and seed. Obliviator (paper-spec) wraps the authors' released code; the only faithfulness check is a synthetic property test (a linear sensitive probe drops after erasure), and its output dimension differs from the input. Its iteration count is the utility dial, and at the owner's matched cost it succeeds on fewer rows than any other arm, so its usable cells are a selected subset: those where a kernel eraser can reach the owner's utility.

Reading	value	declared outcome
Rows at matched utility	LEACE 4800, SPLINCE 3488, LEOPARD 3192, Obliviator 1803 of 4800	reported (O-F)
Usable cells (≥ 20 seeds)	39 (17 constructed)	—
Decision flips, excess (O-A)	2; 0.051 [0.000, 0.128]	not shown
LEACE, same cells	0.051 [0.000, 0.128]	for comparison
Flip direction (O-B)	2 R-release, 0 T-release	reported
Orderings reversed (O-C)	LEACE $12/32$, excess 0.227 [0.116, 0.347]	above 5%
	SPLINCE $10/20$, excess 0.252 [0.128, 0.388]	above 5%
	LEOPARD $0/19$	within 5%
Obliviator−LEACE, R (O-D)	-0.040 [-0.084, -0.018]	no claim
Obliviator−LEACE, T (O-D)	-0.009 [-0.027, 0.014]	(one side only)
Linear audit (O-E)	29 of 36 certified leak > 0.10	reported

O-D claims a difference only when both intervals exclude zero on the same side; here only R's does. The reversals in O-C are what that looks like cell by cell. O-A's null is uninformative rather than negative, since the three other arms barely flip on the same cells either.

D FURTHER CRITERIA FIXED IN ADVANCE (P1, E1′, E2′, E4′)

A companion protocol (`research/PREREGISTRATION_merged_protocol.md`, signed before the extended table was allocated) declared these criteria, their bars and a reporting guard: each is scored on the full table and on the natural-label cells alone, and a verdict that depends on the subset is reported as such. Intervals are cell bootstraps; each criterion must clear a point bar and a lower-bound bar. P1 is the Spearman correlation across cells of the LEACE release's leakage under R and under T, both floor-relative; E1′ the median SPLINCE−LEACE gap at matched utility where nesting is high (with the low-nesting gap required to be null); E2′ whether tuning at matched family size widens that gap; E4′ the median LEACE−LEOPARD (paper-spec) gap in the untuned residual.

	bar	all cells	natural	constructed
cells (median n)		159 (1500)	86 (800)	73 (4000)
P1	≥ 0.55, LB ≥ 0.40	0.465 [0.286, 0.631]	0.509 [0.264, 0.724]	0.394 [0.107, 0.661]
partial on $\log n$	reported	0.451	0.536	0.422
E1′ high	≥ 0.05, LB ≥ 0.02	0.000 [-0.000, 0.001]	0.001	-0.001
E2′	≥ 0.03, LB > 0	0.001 [-0.000, 0.001]	0.001	-0.001
E4′	≥ 0.05, LB ≥ 0.02	0.072 [0.050, 0.103]	0.066 [0.050, 0.114]	0.092 [0.028, 0.123]

No verdict depends on the subset: P1, E1′ and E2′ fail everywhere, and E4′ passes everywhere. On the 108-cell interim table P1 was 0.503 [0.269, 0.701]; the extension narrowed the interval without moving it to the bar. E1′ is a clean null: at matched utility the two linear projections differ in median leakage by 0.000 where the task is nested in the attribute (61 cells), against a bar of 0.05.

E IDENTIFIER MAP

The gates and criteria keep the names they were pre-registered under. They are collected here so the text can stay short.

Name	Where	What it says
F-A	gate, Section 5	common cause: the residual organises geometry and audit error
C1–C2	F-A	Spearman of residual with geometry / audit error, bar 0.60
C3	F-A	the two errors co-move, bar 0.50
C4	F-A	their partial correlation given the residual, bar $< 0.5\times$C3
C5	F-A	the same partial given affinity, bar $> 0.8\times$C3
F-B	gate, Section 5	the residual-aware release rule
D1	F-B	leave-one-panel-out false release < 0.10
D5	F-B	no priced label budget reaches D1
G-C	gate, Section 5	composition drives the tuning gain
P1–P3	G-C	composed-vs-native ratio; video decomposition; dimensionality control
G-D2	gate, Section 5	matched-n control, scored floor-relative
N1′	G-D2	excess declines with n, bar $\rho \leq -0.70$
N2′	G-D2	excess at n=128 reaches 0.15
N3′	G-D2	a control cell with headroom does not move
N4′	G-D2	the excess survives at matched family size, bar 0.10
X1, X2, X4	Section 3	decision flips; method orderings; the linear audit
X3	Appendix P	tuning at matched family size on the designed v2 table
X6	Appendix J	the 30-seed re-run of the composed table
X7, H1–H10	Section 4	the in-sample sweep and its ten declared statements
X8	Appendix B	declared replication of X1, X2, X4 on unseen designed cells
X9, O-A–O-F	Appendix C	Obliviator (paper-spec) as a fourth arm on the extended table
P1, E1′, E2′, E4′	Appendix D	follow-up protocol's criteria, scored on the extended table

F TIES, NOT RELU: THE QUANTISATION EXPERIMENT

Declared in `research/PREREGISTRATION_tie_density.md` before the script existed, with the estimators, splits, attackers and seeds of Section 4 unchanged. Each coordinate is uniformly quantised to b bits over its observed range before any eraser is fitted, so the eraser, the attackers and the utility probe all see the quantised release. Tie density is the mean over coordinates of the largest share of rows holding one value. Verdicts: Q1 pass, Q2 fail, Q3 pass, Q4 fail, Q5 pass; the two failures are discussed in Section 4. Figure 3 shows the sweep.

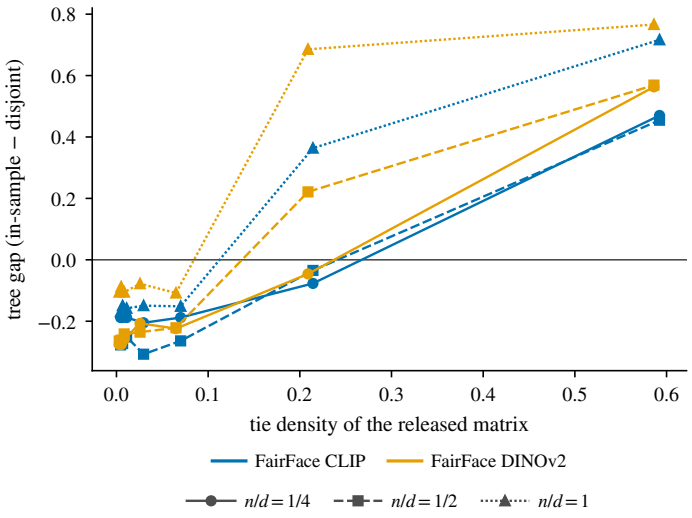

Figure 3: Tree gap (in-sample minus disjoint fitting) against the tie density of the released matrix, for LEACE on two dense embeddings at three ratios. The leak switches on between tie densities 0.070 and 0.214, which is 6 and 4 bits per coordinate.

G DECLARED STATEMENTS FOR THE SWEEP

Every statement declared for the sweep, with its result and its verdict under the declared reading. Three pass outright (H4, H7, H9) and H6 splits, passing on its SAL half and failing on its SPLINCE half, which is the fourth pass counted in the text. The failures of H2, H3 and H8 are discussed in Section 4; H1 and H10 fail only at their declared 10^{-6} tolerances. Declared in `research/PREREGISTRATION_insample_erasure.md`.

ID	Declared statement	Result	Verdict
H1	In-sample LEACE equals class-centroid subtraction when the centred rows have rank $n-1$ (tol. 10^{-6})	holds to 8.5×10^{-14} at $n \le d/2$; 0.029 at $n = d$	fail
H2	Median max-attacker gap (in-sample − disjoint) at $n \le d$ is > 0.05 with positive lower bound	median -0.056 $[-0.168, 0.405]$	fail
H3	Gap falls with n/d on ≥ 4 caches and median gap < 0.02 at $n/d \ge 4$	1 of 5 negative; median -0.030	fail
H4	Tree gap at $n/d \le 1/2$ larger on both ResNet-50 caches than on every dense cache	sparse 0.416–0.682, dense ≤ -0.134	pass
H5	Two-fold cross-fitting matches disjoint fitting ($\lvert\text{median}\rvert < 0.02$) at no utility cost	median nan; utility nan	fail
H6	SAL (unwhitened) median gap < 0.02; SPLINCE (whitened) median gap > 0.05	SAL -0.119; SPLINCE -0.097	pass / fail
H7	In-sample logistic leakage below majority on ≥ 5 of 7 caches at $n \le d$	7 of 7	pass
H8	Remap-aware leakage higher in-sample than disjoint on ≥ 5 caches, median > 0.05	0 of 7; median -0.069	fail
H9	The two families disagree more often under in-sample fitting, mostly R-release/T-refuse	10 of 28 vs 0; all R-release	pass
H10	Train/test class-mean differences antiparallel to 10^{-6} on the binary caches	cosine within 0.0013 of -1	fail

H GATE STUDY FIGURES

Three figures carried over from Section 5: per-cell movement of the residual under attacker tuning, the leave-one-panel-out spread of C1 on the tuned table, and the matched-n control of Gate G-D2.

I RESOLUTION OF C1

Section 6 argues that at n=44 the release rule's bar and the table are mismatched. That argument applies here too, and we would rather make it against ourselves than have it made for us. C1 is a rank correlation over 38 cells, so its large-sample standard error is about 0.11 and a 95% interval is roughly ± 0.21. Rung 1's 0.663 and rung 2's 0.580 are inside one standard error of each other, and both are inside one standard error of the 0.60 bar; the leave-one-panel-out spread in Figure 5 says the same thing without any distributional assumption. So we do *not* claim that the association truly fell when the adversary was tuned.

What we claim is narrower and survives the imprecision. A gate is a decision procedure whose threshold was fixed before the run, so its verdict is well-defined whatever the estimate's precision, and the verdict is do not merge. And the quantity the verdict rests on, per-cell ΔR, is not a cross-cell rank statistic: it is measured within each cell at 30 seeds (Figure 4), and 26 of 44 cells moved past a threshold set in advance. Cell-level movement is what this table can resolve. A precise value for C1 is not.

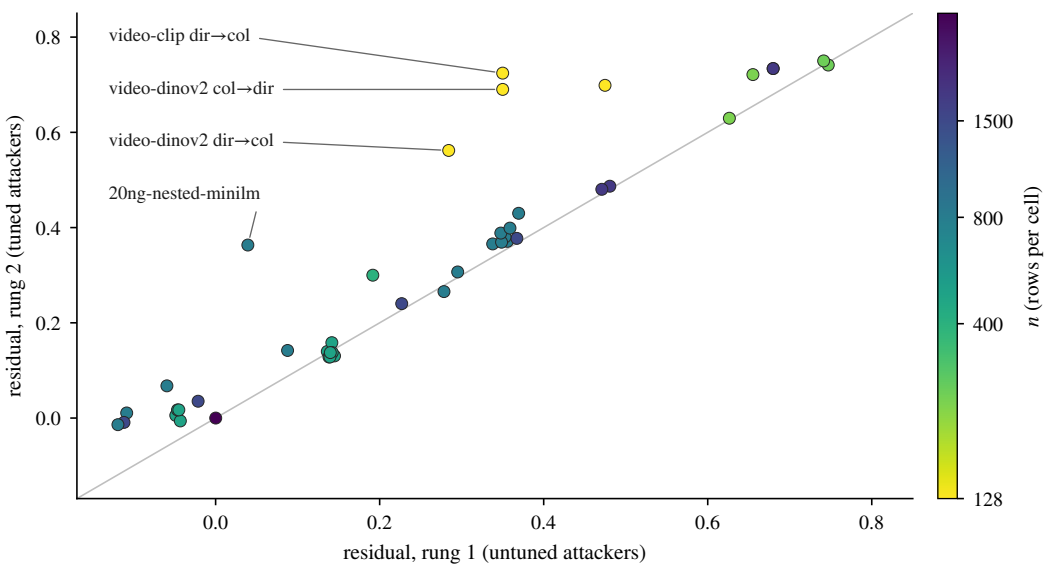

Figure 4: Residual before and after attacker tuning, one point per cell (n=44), coloured by the cell's row count. Points above the diagonal moved under tuning. The four largest movers are labelled: three are video cells at n=128, and `20ng-nested-minilm` is the exception at n=800 discussed below. Panel means quoted in the text pool $|\Delta R|$ within a panel; the points here are individual cells.

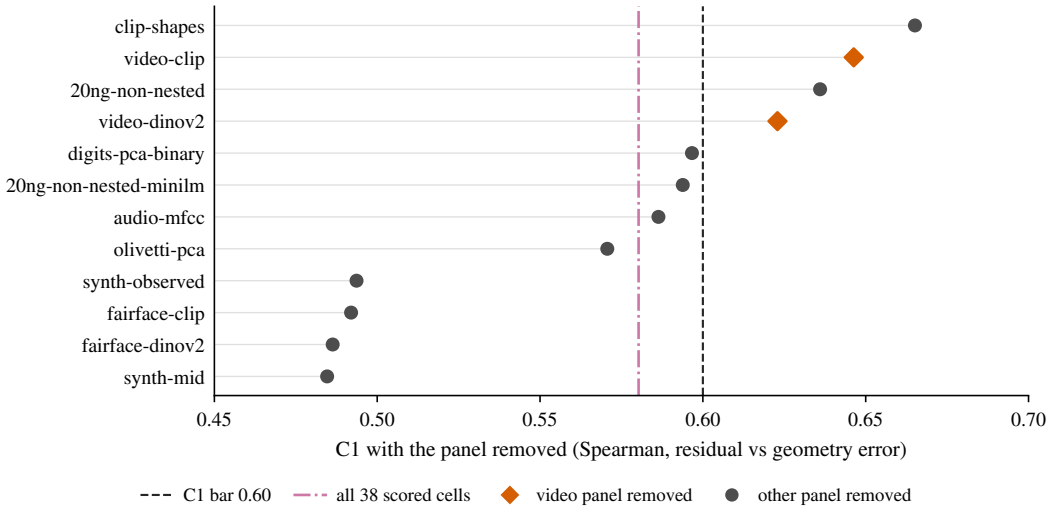

Figure 5: Leave-one-panel-out C1 on the rung-2 table: each row is C1 recomputed with that panel removed, against the 0.60 bar and the value on the full scored table (38 cells, C1 0.580). Dropping either video panel clears the bar (0.646, 0.623); the spread runs 0.485–0.665 over 12 panels.

J GATE G-C SEED COUNT

Gate G-C was first scored on a 3-seed composed table where its pre-registration specified 30. We recorded that deviation with an argument for why more seeds should not change it: P1's permutation test exchanges labels over 11 and 22 cells, so seeds shrink each cell's standard error without adding exchangeable units; the misses were not near-misses; and P2's failure was a reversal of -0.538, not a coin-flip. An argument is not a measurement, so we ran the 30-seed table, declared on 2026-09-19 before the run: 1680 loader-seed measurements on the same frozen cells, with the same estimators and criteria.

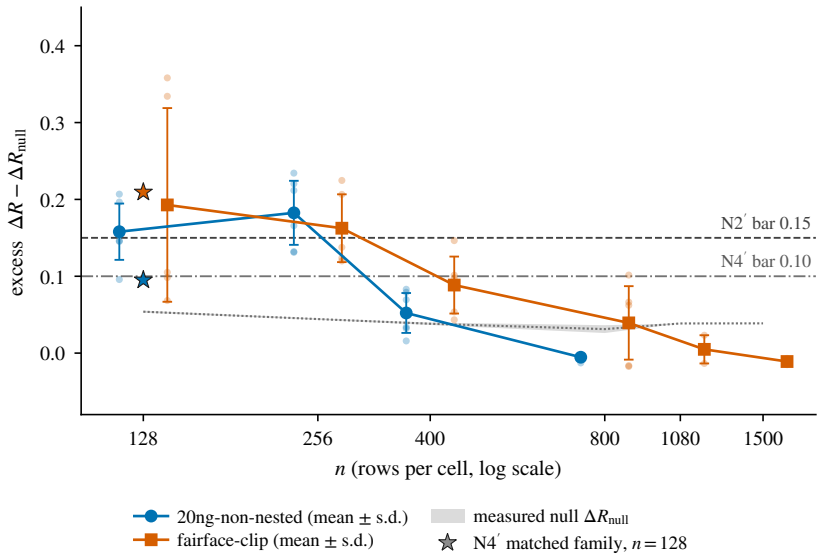

Figure 6: Gate G-D2: floor-corrected excess of tuned over untuned residual against rows per cell, 30 seeds. The grey band is the measured null for a 14-vs-2 family maximum. It is substantial and flat, and removing it leaves the trend. Stars: excess at matched family size, n=128.

The verdict is unchanged and the argument held. P1's ratio moves from $1.91\times$ to $1.72\times$ and the permutation p from 0.12 to 0.26, both far from the $3\times$ bar; the controls pass 1 of 4 and 2 of 5 as before; `all_pass` is false. Both scorings are kept (`results/gate_G_C.json` and `gate_G_C.s30.json`), and the 30-seed one is what Section 5 reports.

K GATE G-D AS FIRST SCORED (WITHDRAWN)

As originally scored against chance, ΔR rises monotonically as n falls on both panels (Spearman ρ of -0.800 on `20ng-non-nested` and -1.000 on `fairface-clip`, against a -0.70 bar). At n=128 the two panels reach mean ΔR of 0.213 and 0.245, against a bar of 0.15 set at half the video effect; at their native 800 rows the same panels move by 0.021 and 0.076. The gate file reports all_pass true. That scoring used chance as the zero point and a control with no headroom; both are withdrawn in the next paragraph.

The panels that carry the rung-2 failure are therefore the panels where the untuned suite was most under-fit, and at matched n the ordinary panels move nearly as much. Attacker tuning appears to buy the most where the fixed suite had the least data to fit, which is a property of the evaluation table, not of concatenated packings.

L WHY THE RUNG-2 ENRICHED RULE IS NOT A PASS

0.091 is 4 of 44 cells, and D1 admits at most 4. Read alone it is a pass. It is not one, for four reasons we state rather than leave to a reader to reconstruct.

1. **It is the minimum of six numbers.** Three rules crossed with two rungs gives six false-release rates, ranging from 0.091 to 0.250, and 0.091 is the smallest. Neither the enriched feature set (added after F-B was scored) nor the rung-2 residual (a robustness check, not the residual D1 was written against) was pre-specified for this test. Two unregistered degrees of freedom, then take the best, is the search D1 was written to prevent.
2. **The interval covers both verdicts.** The rung-2 enriched rate has a paired-bootstrap CI of $[0.023, 0.182]$, or 1 to 8 cells. Its upper end is 0.182, the rate we report as a failure for the combined rule at rung 1. The table cannot separate "clears 0.10" from "misses it by a factor of two."

3. **The whole improvement is one cell.** At n=44, one cell is 2.27 points; the gap between the rung-1 and rung-2 enriched rates is a single cell moving from the wrong side of the decision to the right one.

4. **We already declined this once.** The label-budget curve dips to 0.136 at B=500, and a pre-registration amendment ruled that dip out as an operating point in favour of the asymptote. The same rule applies to a favourable cell of a rule-by-rung grid.

There is a coherent argument for the other reading: that rung 2 is simply the better instrument, so its numbers are the ones that count. We reject it because it cannot be applied selectively. Adopting rung 2 as the operative residual also adopts combined at 0.250, and it adopts a Gate F-A that fails. An instrument that is authoritative where it flatters the rule and advisory where it does not is the failure this paper is about, and we are not going to commit it here.

M FURTHER LIMITS OF THE GATE STUDY

- Which of sample size and *headroom* (the room an untuned attacker leaves above its own residual) drives the concentration. Our residual's zero point is majority-class chance, not the measured floor; Gate G-D2 reports the floor as a sidecar and takes the excess, but the residual the other gates consume is unchanged, and a fifth of the scored cells carry a negative value under it. On the scored table, headroom predicts which cells move once n is partialled out while n does not once headroom is, and inside a subsampling ladder the two cannot be separated at all because subsampling moves both. We hold cells at large n with ample headroom that move and others that do not. Deciding between them needs a floor-relative residual throughout, which is future work and pre-registered as such; we state headroom as a live competing explanation rather than a finding, since as defined it inherits the same zero point.

- A test of C1 and C2 that could plausibly have failed while the calibrator worked. Our own post-hoc check of the scored table finds e_{geom} largely explained by the noise cost, and the residual is what *sets* the noise operating point, so C1 relates a quantity to the cost of acting on it and C2 relates one suite-max on the erased embedding to another. That gives a duller reading of this paper's headline than the one we advance: C1 may fall between rungs partly because rung 2 is not the suite that set the operating point, so a shared instrument weakens. We report it because a paper about instrument choice does not get to leave its own primary criteria unexamined; specifying an instrument-independent primary is the first item of the follow-on protocol.

- A mid-band affinity fill on unsupervised semantic encoders. The real mid-band cells we hold are two supervised ResNet panels, FairFace (α≈0.77) and UTKFace (α≈0.69), both kept outside the F-A cap; between them and the saturated ceiling the band is synthetic. Every claim we make about the high-affinity stratum therefore rests on cells whose geometry we constructed. Filling that band on CLIP, DINOv2 or SigLIP is the first thing we would run next, and we did not run it for this paper.

N IN-SAMPLE ERASURE AT $d \geq n$, WITH CONTROLS (POST HOC)

Sensitive label: race. Entries are raw test accuracy against the chance column, not the leakage-above-majority used in the main text. LEACE is fitted on the n rows released (in-sample), on a disjoint half (disjoint fit), or in-sample after PCA to 256 dimensions; attackers are logistic regression and a depth-4 gradient-boosted tree, scored on a 70/30 split of the released rows. The UTKFace CLIP row ($d < n$) and the PCA-256 columns are controls. Post hoc; `scripts/diag_insample_erasure.py`.

Embedding	d	n	seed	chance	in-sample logit	in-sample tree	disjoint fit logit	disjoint fit tree	PCA-256 logit	PCA-256 tree
FairFace ResNet-50	2048	1000	0	0.213	0.003	1.000	0.280	0.277	0.027	0.100
FairFace ResNet-50	2048	1000	1	0.183	0.010	1.000	0.277	0.303	0.020	0.090
FairFace ResNet-50	2048	1000	2	0.203	0.003	1.000	0.250	0.270	0.017	0.103
UTKFace CLIP	512	1000	0	0.427	0.140	0.550	0.447	0.563	0.170	0.410
UTKFace ResNet-50	2048	1000	0	0.427	0.207	0.993	0.493	0.543	0.243	0.377
UTKFace ResNet-50	2048	1000	1	0.431	0.223	0.993	0.477	0.527	0.250	0.407
UTKFace ResNet-50	2048	1000	2	0.447	0.260	1.000	0.477	0.563	0.313	0.400

O DESIGNED-TABLE SENSITIVITIES

Table 3: Method orderings under families R and T. An ordering is resolved under a family when the paired seed-level difference exceeds 2 standard errors; cells need ≥ 20 seeds matched for both methods. "Reversed" is resolved under both families with opposite signs.

Pair	cells	same order	reversed	resolved in one	tied in both
LEACE–SPLINCE	72	1	1	15	55
LEACE–LEOPARD (paper-spec)	61	37	1	9	14
SPLINCE–LEOPARD (paper-spec)	50	28	1	8	13

Excess release-decision disagreement (cross-family minus within-family, split-seed null) with cell-bootstrap 95% intervals. All rows at t=0.05 unless stated. Declared in `research/PREREGISTRATION_instrument_decisions.md`; the last row is post hoc.

Subset	excess	95% CI	cells
All, t=0.02	0.087	[0.050, 0.128]	109
All, t=0.05	0.076	[0.043, 0.116]	109
All, t=0.10	0.152	[0.100, 0.212]	109
LEACE	0.119	[0.064, 0.183]	109
SPLINCE	0.042	[0.000, 0.097]	72
LEOPARD (paper-spec)	0.039	[0.011, 0.075]	61
Affinity stratum S1	0.027	[0.000, 0.077]	32
Affinity stratum S2	0.137	[0.036, 0.265]	30
Affinity stratum S3	0.057	[0.012, 0.127]	19
Affinity stratum S4	0.173	[0.054, 0.346]	18
Affinity stratum S5	0.000	[0.000, 0.000]	10
Nesting high	0.067	[0.029, 0.115]	69
Nesting low	0.240	[0.094, 0.476]	16
Leave-one-panel-out (min / max point)	0.052 / 0.090	LB > 0 in all	21 panels
Without ResNet-50 panels (post hoc)	0.045	[0.016, 0.082]	97

P TUNING AT MATCHED FAMILY SIZE (X3, DESCRIPTIVE)

The designed v2 table records, for each cell–method–seed, leakage under a fixed size-2 attacker family and under a size-2 family chosen as the best of two from the tuned grid (both floor-corrected). That is the designed-table relative of Gate G-D2's N4′: does tuning at matched family size flip the release decision? Completion order is the allocator's, not a random sample, so the analysis is descriptive, states the panel skew, and is not pooled with Section 3. It was first scored on the 35 cells complete at a declared cutoff; the numbers below are the same unchanged analysis re-run on all 160 cells once the sweep finished ($2026 - 09 - 25$), which removes the CelebA skew of the interim score (65% of cell–arms then, 21% now).

X3 on `x8_v2_rows.json` (160 cells): release decisions under a fixed size-2 attacker family versus a matched size-2 tuned family (both floor-corrected). Descriptive only; completion order is the allocator's. Scored 2026-09-25. Primary t=0.05.

Quantity	value	95% CI	n
Cell–arms ($\geq$ 20 seeds)	350	—	160 cells
Full flips at t=0.05	294/350 (84.0%)	—	—
Excess (cross − within)	0.826	[0.769, 0.878]	160
Fixed releases / matched refuses	294	—	—
Matched releases / fixed refuses	0	—	—
CelebA CLIP share of cell–arms	21%	—	—
Leave-one-panel-out excess (min / max)	0.777 / 0.858	LB $\leq$ 0: no	—

Panel counts among scored cell–arms: celeba (72), synth-designed (44), fairface-clip (27), utkface-clip (27), fairface-dinov2 (25), utkface-siglip (19), 20ng-non-nested (17), utkface-dinov2 (16).

The fixed pair here is two logistic regressions and the matched pair is the best two of the same six-member family on a selection split, so this compares tuning *within* one family rather than a change of attacker class. Under in-sample fitting the linear members are exactly the ones Section 4 shows to be depressed, and that is the direction observed: every flip is a fixed-family release that the matched family refuses, never the reverse.

At t=0.05, 294 of 350 cell–method pairs flip (84.0%), with split-seed excess 0.826 [0.769, 0.878], all of them the fixed pair releasing what the tuned pair refuses. CelebA CLIP accounts for 21% of the scored cell–arms; leave-one-panel-out excess ranges 0.777–0.858 with the lower bound above zero throughout. Within a single family, then, an owner who runs two default logistic probes releases what a best-of-two search refuses on most of the table.