SATURATED SCREENING, CONCENTRATED COST: WHAT A FULL-STACK ABLATION TABLE READS IN A REDUNDANT REFUSAL STACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety teams assemble refusal stacks — released guard models alongside the assistant's own refusal behaviour — composed so that any one component can block, then decide what the stack is worth by ablating it: switch each component off in turn and see what breaks. We ask what that table measures. We assemble 9 refusal sources (8 released guard models from 5 vendors, plus the assistant's own refusals), run each guard model at the operating point its vendor ships, and enumerate every one of the 512 subsets over 3937 prompts from 4 suites, so each source's exact Shapley value sits beside its leave-one-out score with no coalition sampling (attribution figures: main suite). The harmful side is saturated: the full stack passes no harmful prompt on any of the 3 suites that contain harmful prompts, and not one has a sole blocker, a mean of 8.56 [8.46, 8.64] of the 9 sources blocking each StrongREJECT prompt. Switching any single source off therefore changes not one harmful decision at these shipped operating points, so every full-stack leave-one-out number here varies only through over-refusal. A zero full-stack removal marginal need not imply zero coalition-averaged contribution: on the harmful side the exact Shapley values sum to -1.0000 while the leave-one-out entries sum to 0.0000, and ablation sends 5 of the 9 to exactly +0.000 while those same sources carry 71.5% [65.0, 76.5] of $\sum_i |\phi_i|$ under that normalisation, and the loss-minimising subset is drawn from them (WildGuard-7B alone in the unrestricted game, at 11.30 [7.84, 18.51]$\times$ lower loss than the full stack, though it passes 20 harmful prompts). So a one-shot rule deleting that column's zeros removes the sources that carry most of that credit, and the cost is not hypothetical: the union blocks 98.6% [98.0, 99.2] of OR-Bench hard benign prompts. Two things follow cheaply. Calibrating the guard models to a 3% benign false-positive rate on the XSTest-safe pool cuts the union's benign block rate 3.9$\times$ (2.57–7.20$\times$ refitted on 4 benign pools) and cuts that same $\sum_i |\phi_i|$ zeroed-mass statistic 4.4$\times$; it also returns 8 harmful prompts to a sole blocker. That is only a start, as 98.4% of the harmful prompts are still blocked by two or more sources, but it shows the saturation is a property of the operating points these guards ship with, not of any-one-blocks composition itself. Nor is the support collapse the 9-way union's: by coalition size 4, 53.6% of (coalition, member) pairs already read exactly +0.000 on the harmful side (46.7–60.0% with one guard model per vendor). Audit a redundant stack for what it costs: here no single removal changes a harmful decision, so full-stack ablation has no harmful-side support.

1 INTRODUCTION

A team assembles a refusal stack: 8 released guard models from 5 vendors, plus the assistant's own refusal behaviour, wired so that a block from any one of them blocks the request (§2.1 names and cites all 9). It refuses far too much: the union blocks 98.6% [98.0, 99.2] of OR-Bench hard benign prompts (Cui et al., 2025) and 79.2% [75.6, 82.8] of PHTest's plainly harmless tier (An et al., 2024), while XSTest-safe (Röttger et al., 2024), the benchmark the field reaches for to bound over-refusal, is the *most* forgiving of our four benign sources at 57.6% [51.6, 63.6] (§3.1).

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.

So, do we need all 9? The standard answer is an ablation table (switch each guard off in turn and record the change), and on this stack it cannot answer on the harmful side. **The harmful side is saturated.** The full stack blocks every harmful prompt of all four harmful sources (§2.1 names them), so $H(N) = 0.0000$ (per-source rates in Appendix B), and none of them anywhere has a *sole* blocker (§3.2). Since a guard changes the system's decision only where it blocks alone, removing any single guard changes *zero* harmful decisions, so at these operating points the harmful term of every full-stack leave-one-out is arithmetically zero. That follows from how these guards ship: matched to a common benign false-positive rate, 8 harmful prompts regain a sole blocker against 0 (Table 1).

That is the paper's central claim, and it is a claim about the instrument: **the full-stack ablation table of a saturated refusal stack varies only through cost, whatever the team running it believes it reads.** Over-refusal is where one guard *can* be the sole blocker, so the same runs answer sharply there (§3.2).

On the readable side a dead-weight reading of the table still misleads, because cost-reducing guards overlap too. Over all 512 coalitions of the main suite, the rows for {L, S, L4, W, L1} read +0.000 exactly and the natural reading is dead weight; yet those 5 hold 71.5% [65.0, 76.5] of $\sum_i |\phi_i|$, and the loss-minimising subset {W} is drawn entirely from them at 11.30 [7.84, 18.51]× lower loss than all 9. That credit share is convention-dependent (§4.3); that the minimiser sits inside the table's zeros is a property of this fixed game, tested where the game changes (§5.1, Appendix F). That minimiser is also selected on the data its ratio is computed from, so we price the winner's curse: held out the gap is 10.41 [5.18, 17.45]× against 11.30× (§4.2).

Three cheap controls diagnose or bypass the failure, none needing 2^n runs. Matching each guard to a common benign false-positive rate rather than a common threshold collapses the gap from 11.30× to 2.50× at 3%, and the guards it zeroes become G, G8. The minimiser is no longer among them. Held out over 200 splits the gap is 2.79 [2.06, 4.29]×, *larger*, so the advice survives it (Appendix G). It is the only control that moves both failures, cutting the union's benign rate 3.9× — still far above target — while returning part of the harmful side to view (§5.1). Two more need no enumeration: greedy backward elimination reaches the exact minimiser (§4.2), and leave-one-*in* resolves the 5-way zero tie ablation cannot (§5.2).

Contributions. *One:* we operationalise the classical criticality condition (Birnbaum, 1969) as a support audit for compound refusal systems, keeping apart the full-stack removal marginal, coalition-averaged attribution, and subset selection. *Two:* 8 released guards at their shipped operating points lose harmful-side removal support as redundancy grows: 53.6% of removal pairs are already zero at $m = 4$ (Table 2), 46.7–60.0% with one guard model per vendor, and 20.0–28.3% once those rosters are also calibrated. *Three:* we locate the boundary of that regime (matched calibration, other composition rules, convention and detector stress, recapture diagnostics) and introduce no new attribution estimator.

2 Setup, and what is already known

2.1 The stack, the value function, and the data

The audited system is 8 released guard models from 5 vendors, which *we* assembled from public checkpoints: not anyone's production system and deliberately untuned, since only the empty coalition scores strictly worse than all 9, and 31 others tie it. It is worth enumerating because every guard is a released checkpoint at its shipped operating point and the remaining player is the assistant's own refusal behaviour, not removable while this generator is fixed (the lattice removes it on paper, discounting its detected refusals): `granite-guardian-3.0-2b` (G) and `granite-guardian-3.0-8b` (G8) from IBM (Padhi et al., 2025), `Llama-Guard-3-8B` (L), `Llama-Guard-3-1B` (L1) and `Llama-Guard-4-12B` (L4) from Meta (Inan et al., 2023; Grattafiori et al., 2024; Meta, 2024; 2025), `ShieldGemma-2B` (M) from Google (Zeng et al., 2024), `Qwen3Guard-Gen-4B` (Q) from Alibaba (Zhao et al., 2025), `WildGuard-7B` (W) from AI2 (Han et al., 2024), and the self-refusal behaviour (S) of the `Llama-3.1-8B-Instruct` generator itself (Grattafiori et al., 2024), detected by a keyword matcher whose error rates are measured rather than assumed from the released adjudication records, and applied to the lattice as all 20 adjudicated corrections (§6). The guards compose by *any-one-blocks* (a prompt is refused if at least one member of the coalition refuses it), which is why such stacks are redundant by construction. The redundancy is deliberate, so that no

single guard's failure becomes the system's. Each guard decides at its vendor's documented default operating point under that vendor's own output convention — the choice we interrogate hardest (§5.1).

For a coalition $C \subseteq N$ the value function, used everywhere, is

$$v(C) \;=\; H(C) \;+\; \lambda\, F(C), \tag{1}$$

where $H(C)$ is the fraction of harmful prompts the coalition passes and $F(C)$ the fraction of benign prompts it blocks; we report at equal weighting, $\lambda = 1$. Both count refusal *decisions*, not what a passed prompt goes on to produce. A removal sees less than it looks. Under k-of-n blocking with k fixed, removing i changes a prompt only when i blocks it and exactly k members do, so under any-one-blocks its support is the prompts i blocks *alone* — the classical criticality condition (Birnbaum, 1969). Three questions follow: full-stack leave-one-out asks what changes when a guard leaves the coalition being run, the Shapley value averages that marginal over coalition contexts, and subset search asks which coalition minimises the loss. Redundancy can drive the first to zero without zeroing the other two, so a zero here is the correct local marginal; reading it as credit is what fails. It is a *loss*, so a negative Shapley value means the player *reduces* loss and is useful; every sign reads that way. The two terms behave completely differently, and that asymmetry is the paper (§3.2). At these operating points H is saturated at 0.0000 and locally constant under every removal, while F stands at 0.5760 and moves under only 4 of the 9 removals.

All 512 coalitions were enumerated, not simulated, on four prompt suites drawn from five benchmarks, each answering a different objection. The main suite is StrongREJECT (Souly et al., 2024) plus XSTest (Röttger et al., 2024): 313 harmful items and XSTest's contrastive pairs (250 safe, 200 unsafe). JailbreakBench (Chao et al., 2024) adds 100 harmful and 100 benign prompts matched by category and index; OR-Bench (Cui et al., 2025) its hard-1k benign-but-alarming split (1319) against a toxic control (655); PHTest (An et al., 2024) its harmless and controversial tiers (500, 500), benign throughout, so its lattice is the benign side alone (Appendix A). Uncertainty on every lattice quantity comes from a cluster bootstrap, $B = 6000$, over 513 clusters of the main suite. An XSTest twin group resamples as a unit, and on JBB each matched pair does (Cameron et al., 2008); Appendix B gives the construction. A second system (§4.4) is a five-gate QA abstention cascade over Korean public-health data, whose full 2^5 lattice we also enumerate.

2.2 What is already known, and what it does not cover

The cooperative-game apparatus is classical and we use it as it stands: the OR simple game as the dual of a unanimity game (Shapley, 1953); the dividends and interaction indices that locate missing credit (Harsanyi, 1963; Owen, 1972; Grabisch & Roubens, 1999; Sundararajan et al., 2020); removal-based attribution's dependence on the reference distribution, and redundant components splitting and masking each other's credit (Covert et al., 2021; Lundberg & Lee, 2017); and causal responsibility under redundant causes (Chockler & Halpern, 2004), used in §3.2.

For convex and concave games the ordering between a player's marginal contribution to the grand coalition and its Shapley value is fixed by the curvature of v (Shapley, 1971). It does not license attributing our gap to that dichotomy. Our loss is a difference of two submodular set functions (coverage on each side), and testing the curvature inequalities directly on $H + \lambda F$, over the nested triples the definition ranges over, finds substantial violations in *both* directions, so the game is neither convex nor concave. The ordering $\phi_i \leq \mathrm{LOO}_i$ holds for all 9 players at $\lambda{=}1$ as a *measurement on this game*, not a corollary.

Concurrent MeClear (Yang et al., 2026a) also resolves redundant leave-one-out masking with cooperative attribution, on agent memory. We ask instead whether released refusal guards at shipped operating points lose harmful-side support for the full-stack removal marginal, how early that happens as the stack grows, whether the benign side collapses too, and whether calibration reverses it. The masking mechanism is not our novelty; the measured regime and its boundary are.

Table 1: Saturation is a property of the shipped operating points, not of the composition rule. Matched to a common benign false-positive rate the guards leave 8 harmful prompts with exactly one blocker, against 0 at the vendor defaults, while the union's benign block rate falls 3.9×, though it stays far above the target. Benign pool XSTest-safe; targets fitted in sample (held out, Table 31); S exposes no score (§5.1).

operating point	$H(N)$	sole-blocked	min $\lvert B\rvert$	union benign rate
vendor default	0.0000	0	3	57.6%
matched benign FPR 3%	0.0000	8	1	14.8%
matched benign FPR 5%	0.0000	4	1	20.8%
matched benign FPR 10%	0.0000	1	1	34.8%

3 What the stack does: severe over-refusal, saturated harmful coverage

3.1 Over-refusal is severe, and XSTest is the forgiving case

XSTest's safe half is built to look alarming, said to exaggerate a stack's cost; our four benign sources answer in the wrong direction: the union blocks 98.6% [98.0, 99.2] of OR-Bench hard-1k (Cui et al., 2025), 96.4% [94.6, 98.0] of PHTest controversial and 94.0% [89.0, 98.0] of category-matched JailbreakBench benign prompts against 57.6% [51.6, 63.6] of XSTest-safe, and 79.2% [75.6, 82.8] of PHTest's plainly harmless tier.

A stack assembled this way rejects most benign items on every source we tried, and the failure is not one bad component's — it is the union of the 9 vendor-default and self-refusal decisions.

3.2 Blocker multiplicity: harmful coverage is redundant, cost is concentrated

The asymmetry §4.1 reads off the lattice is measurable directly. Let b count the guards blocking a prompt. A block survives removing i exactly when $b \geq 2$, so leave-one-out sees a guard only where it is the sole blocker (Table 21). Harmful prompts are blocked by 8.56 [8.46, 8.64] of 9 guards on StrongREJECT and 7.97 [7.74, 8.18] on XSTest-unsafe, with $b \geq 2$ for 100.0% and 100.0% of them and the same saturation on the other two harmful sources. Over-refused benign prompts are blocked by 2.34 [2.11, 2.58] guards on XSTest-safe and by 3.98, 4.72 and 2.79 on the other three, and the share with $b \geq 2$ falls to 63.9 [55.9, 71.7]% on XSTest-safe and 71.5% on PHTest harmless. A substantial fraction of over-refusals thus have a sole blocker and are visible to an ablation. On JBB, where the benign item shares a category and an index with the harmful one it is paired against, the paired difference in b is 4.27 [3.80, 4.75] and excludes zero, so the asymmetry survives matching by category and index and is not aggregate topic mix alone.

Responsibility. The Chockler–Halpern construction (Chockler & Halpern, 2004) gives a guard $1/b$ for a prompt it blocks with b blockers in all, averaged over each side of the loss. Under any-one-blocks it is exactly the normalised Shapley value of that side, so it restates the split legibly rather than confirming it (Appendix B). Harmful shares are near-uniform, within 1.35× of each other and close to the 0.1111 a symmetric stack would give; benign shares run to 57.7× between granite-guardian-3.0-2b and WildGuard-7B. The responsibility and Shapley views of the harmful side of a saturated stack read a near-symmetry with no clear ranking.

Saturation is a property of the shipped operating points. Every multiplicity above is measured at the vendor defaults, so the redundancy belongs to them, not to any-one-blocks in general. Re-thresholding to a common benign false-positive rate partly desaturates the harmful side: sole blockers reappear at all three targets while the stack still passes 0 harmful prompts at 3% (Table 1).

Table 2: The lattice read by coalition *size* m, every entry aggregated over all $\binom{9}{m}$ coalitions of that size on the main suite. The harmful side's removal support collapses from $m = 2$ on, and 53.6% of pairs read zero by $m = 4$, so the saturation is not the maximal union's doing. *best F, H*=0 is the lowest benign block rate at that size with no harmful prompt passed.

m	1	2	3	4	5	6	7	8	9
median H	0.0390	0.0088	0.0019	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
best F, H=0	—	0.180	0.180	0.180	0.196	0.212	0.376	0.468	0.576
≥ 2 blockers	0.0%	88.3%	96.5%	98.5%	99.3%	99.7%	99.9%	100.0%	100.0%
$\mathrm{LOO}_i^H = 0$	0.0%	9.7%	32.1%	53.6%	73.0%	87.7%	96.4%	100.0%	100.0%

Table 3: Coalitions of the 9-guard stack on the main suite (313+200 harmful items, 250 benign). H, F and $v = H + \lambda F$ at $\lambda = 1$ as defined in §2.1, lower better; ϕ_i exact Shapley, $\mathrm{LOO}_i = v(N) - v(N \setminus \{i\})$ the ablation score, negative reducing loss. Intervals are cluster-bootstrap percentiles ($B = 6000$, 513 clusters); the $\lambda = 1$ minimiser is the single-guard row {W}. Full lattice in Appendix A.

Coalition	H	F	v	ϕ_i	LOO_i
{G} granite-guardian-3.0-2b	0.0019	0.3960	0.3979	0.084 [0.046, 0.126]	0.108 [0.071, 0.148]
{L} Llama-Guard-3-8B	0.0877	0.0280	0.1157	-0.102 [-0.106, -0.098]	0.000 [0.000, 0.000]
{S} self-refusal	0.0331	0.0600	0.0931	-0.105 [-0.112, -0.097]	0.000 [0.000, 0.000]
{M} ShieldGemma-2B	0.2242	0.2720	0.4962	0.046 [0.015, 0.082]	0.056 [0.030, 0.086]
{G8} granite-guardian-3.0-8b	0.0039	0.3080	0.3119	0.001 [-0.027, 0.031]	0.028 [0.008, 0.053]
{Q} Qwen3Guard-Gen-4B	0.0175	0.1400	0.1575	-0.065 [-0.084, -0.043]	0.016 [0.004, 0.033]
{L4} Llama-Guard-4-12B	0.1404	0.0680	0.2084	-0.084 [-0.091, -0.076]	0.000 [0.000, 0.000]
{W} WildGuard-7B	0.0390	0.0120	0.0510	-0.112 [-0.117, -0.106]	0.000 [0.000, 0.000]
{L1} Llama-Guard-3-1B	0.1248	0.0640	0.1888	-0.087 [-0.094, -0.080]	0.000 [0.000, 0.000]
full stack (assembled here)	0.0000	0.5760	0.5760		

4 THE LATTICE, AND WHAT ITS ABLATION COLUMN READS

4.1 THE MEASUREMENT, AND THE DEGENERACY INSIDE IT

Neither measurement above needed a lattice; this one is built from real runs (all 512 subsets enumerated on the main suite, every self-refusal decision carrying the 20 adjudicated corrections of Appendix C) to show what an ablation table over this stack reads. Nothing below rests on the composition being well chosen (§2.1).

Read the harmful column of Table 3 before the attribution columns. The full stack sits at $H = 0.0000$ and passes nothing. By §3.2 no harmful prompt in this study has a sole blocker at these guards' shipped operating points, though single guards pass plenty alone (M passes 0.2242). So $H(N) = H(N \setminus \{i\})$ exactly, for every i, and the whole LOO_i column is λ times a difference in over-refusal rates. ***At these vendor-default lattices the harmful term of every full-stack leave-one-out is zero.*** That is arithmetic rather than estimation, and it holds on every any-one-blocks lattice we enumerated at those points, though not under the matched calibration of §5.1. Nor does it need 9 guards. Read by coalition size, 53.6% of (coalition, member) pairs are already exactly zero on the harmful side at $m = 4$, where 98.5% of blocked harmful prompts carry two or more blockers (Table 2); forcing one guard model per vendor (S kept as its own player) leaves 46.7–60.0% at $m = 4$ across all 6 rosters, and 20.0–28.3% with those rosters also at the matched 3% rate (Appendix G). That is counting: a prompt with $b \geq 7$ blockers, 91.0% of the harmful prompts here, supports no size-4 removal at all (Appendix A).

How large is the blind spot? Unusually, the question needs no credit convention. Since v is linear, $\phi_i(v) = \phi_i(H) + \lambda\phi_i(F)$, and efficiency makes the harmful halves sum to $H(N) - H(\emptyset) = -1.0000$, the stack's whole harmful-side value, while the harmful leave-one-out entries sum to 0.0000. Shapley distributes all of a quantity ablation reports none of, independent of how Table 20 normalises the shares (Appendix B, Table 17).

On the cost side the table is readable in principle and misread in practice, because cost-reducing guards overlap each other too. For {L, S, L4, W, L1} ablation returns +0.000 while their Shapley intervals are the 5 most negative and all 5 exclude zero on the helpful side. The ordering $\phi_i \leq \mathrm{LOO}_i$ holds for all 9 players at λ=1, measured on this game rather than derived from curvature, which ours lacks (§2.2).

One rule governs sign disagreements throughout (a disagreement needs both intervals to exclude zero), giving the strict set {Q} with {G8} unsupported rather than contradicted; exact zeros are counted apart from it (Appendix D).

4.2 One-shot pruning on the zeros fails; iterative backward search does not

A one-shot rule that reads an exact zero as a deletion candidate removes {L, S, L4, W, L1}. The subset {W}, built entirely out of guards ablation scores at zero, attains 0.051 against the full stack's 0.576 (11.30 [7.84, 18.51]×), but it buys that by passing 0.0390 of the harmful side, most of its own loss. A subset that passes harmful prompts is not one we would call deployable, and the lattice answers that without a further run: 346 coalitions hold H at 0.0000, and 4 of them tie at the lowest benign rate 0.1800 against 0.5760, the smallest {S,Q}. It is a candidate to validate, not to deploy: refitted on half the clusters it stays harm-free held out in only 90/200 splits, under half; {W} is what the $\lambda = 1$ loss says. It is the exact minimiser for $\lambda \in [0.585, \infty)$, and above 0.019 the minimiser is always a *subset of the zeroed guards* (100.0% of resamples; {W} itself optimal in 81.3% of them at $\lambda = 1$ and 97.7% at $\lambda = 2$; at $\lambda = 0.25$ the minimiser is {S,W} instead, reselected in 54.3% of resamples, and {W} in 0.5%). The window above 0.585 is unbounded: no further set takes over as the benign weight grows.

That gap is a ratio whose minimising subset is selected on the clusters the ratio is computed from — the sharpest statistical objection to this section — so we price it. Choosing the argmin on half the clusters and evaluating its gap on the other half, over 2000 splits, gives 10.41 [5.18, 17.45]× against 11.30× in sample: the curse costs 7.9% of the gap, the intervals overlap heavily, the halves agree on the subset in 60.7% of splits and {W} wins 80.6% (Table 31).

The one-shot reading fails; search does not. Greedy backward elimination reaches the exact global minimiser on every enumerated suite and every λ: on the main suite at $\lambda = 1$ it keeps {W}; forward selection does not.

4.3 How much credit, and under which convention

"The zeroed guards hold most of the credit" is denominator-dependent. We keep Shapley over Banzhaf because it is *efficient*, the values summing to $v(N) - v(\emptyset)$, but mixed signs leave no canonical denominator for a share: the printed 71.5% [65.0, 76.5] divides by $\sum_i |\phi_i|$, and under the efficiency denominator the ratio exceeds 100%. Two restrictions push it far the other way. With S held fixed — the only counterfactual a deployment can build — the zeroed set is {L, L4, W, L1} at 3.6% with 0 disagreements, and under Banzhaf (Banzhaf, 1965; Dubey & Shapley, 1979) 4.4% with 0, a spread several times wider than the bootstrap interval on any one of them. *The credit mass is not a stable quantity, and any single figure for it, ours included, is a choice of convention*; we print them all and rest no conclusion on the number. Neither the zeroed set nor the minimiser moves with the weighting. The game fixes both and ϕ enters neither, so these conventions test only how much credit those rows get. The relation is tested where the game changes. With S held in, the minimiser is S plus {W}, 7.59× better than all 9; matched calibration and Q's stricter convention break it (§5.1, Appendix F).

Nor are the players exchangeable: Owen's value (Owen, 1977) over the vendor partition (S a singleton) gives the zeroed set 63.9% of $\sum_i |\phi_i|$ and *reorders* 6 of 9 without a sign change (Appendix G), so the credit *ordering* is a convention too.

4.4 Robustness: other suites, other rosters

Were the cost-side result an artefact of XSTest's contrastive construction it would not survive a change of benign source, so we enumerated the lattice again on JailbreakBench and OR-Bench, each with its own harmful control. Both replicate, saturation included: the stack blocks every harmful item on both

and is beaten by a subset drawn entirely from the guards ablation zeroes, by 3.76× on JBB and 2.77× on OR-Bench, with zeroed sets {L, S, G8, L4, W, L1} at 58.6% of $\sum_i |\phi_i|$ and {L, L4} at 35.2% against {L, S, L4, W, L1} and 71.5% [65.0, 76.5] on the main suite (§4.3). The effect is weakest on OR-Bench and we report that rather than the average. L, L4 score exactly zero on all 3, and every suite's minimiser is drawn from its own zeroed set (3 of 3). But *which* subset that is depends on the benchmark ({W} against {L}), so a team pruning to one suite's answer carries a guard another calls surplus.

The other objection is that one over-eager guard manufactures the result, and the suspect is named in advance: granite-guardian-3.0-2b alone blocks 0.3960 of the benign half and carries the largest benign responsibility share. We re-enumerated with each guard deleted in turn, and again with the roster cut to one guard per bloc — 10 further complete lattices (Appendix G, Table 27). None dissolves it: across the 9 deletions the share stays between 62.4% and 74.0%, the gap never falls below 7.05×, and one guard per bloc gives 66.5% at 10.75×. What is robust is that floor and not an invariance — only M raises the share, and the minimiser comes from that lattice's own zeroed guards in 9 of 10 rebuilds rather than all of them (Appendix G quantifies both). The effect is not one member of this roster's; its size is the operating points' (§5.1).

Partial replication in a system that is not a guard stack. Redundant veto composition is not specific to safety filters, so we repeated the enumeration on a five-gate abstention cascade over Korean public-health question answering (rule-based gates over structured data, sharing neither training corpus nor architecture, composed by the same rule). What replicates is the zeroing, and only that. Three of the five gates have every abstention recaptured by a gate that remains, so their leave-one-out is exactly zero while all three carry nonzero Shapley value, out of sample too, on 150 held-out questions (selective risk 0.267 without abstention, 0.128 with). Only the zeroing half replicates: the cascade's own minimiser contains a gate whose leave-one-out is nonzero, so the best-subset half has no counterpart here (Appendix E, Table 24).

5 Three cheap controls, and what a bounded audit budget can conclude

5.1 Operating points: a common threshold is not a common operating point

We store the decision-token log-probability behind every decision, so re-thresholding needs no re-run. A sweep of five common settings of p_{unsafe} leaves the gap intact (3.39–12.09×, Appendix D) but preserves the calibration mismatch it is meant to test, the guards' scores sharing no scale.

The control the lattice section should have run matches the guards on the axis the loss prices: a common benign false-positive rate, each thresholded to block at most that share of the benign pool. At 3% the full stack's loss falls from 0.576 to 0.148 while the best subset stays {W}, its loss moving from 0.051 to 0.059. The gap collapses from 11.30× to 2.50×; ablation now zeroes G, G8, all 2 of which it scored as costly at the vendor defaults. At 5% the gap is 2.76× with no guard zeroed, at 10% 3.74× zeroing G8. Is the minimiser still drawn from the guards ablation zeroes, at any target? no: the best-subset half of §4.2 holds at the vendor defaults, not under matched calibration. The collapse is almost all the stack getting better. The shipped vendor defaults drive much of the assembled stack's 0.576; redundancy still compounds the matched components to a 14.8% union rate. The same re-thresholding partly desaturates the harmful side (§3.2, Table 1).

Three caveats, the first half-discharged. *The thresholds were fitted in sample, and on one benign pool*; refitted on half the benign clusters and evaluated on the other half over 200 splits the gap is 2.79 [2.06, 4.29]× against 2.50×, *larger* held out, though the loss levels behind it move the other way (Appendix G, Table 31). Refitted on each of the 4 benign pools the cut ranges 2.57–7.20× and only 1 of the 4 pools leaves $H(N)$ at exactly zero (Table 29). *S exposes no score* and enters every matched-FPR lattice at its keyword decision. And *the exact-zero pathology shrinks rather than dissolving*: ablation still zeroes a guard at 2 of the 3 targets, and at every target leave-one-out still has the opposite raw sign to ϕ on 6 to 8 of 9 guards; these re-thresholded lattices carry no intervals, so those are upper bounds on strict disagreements (Appendix D).

5.2 THREE BUDGETS, AND A FOURTH MEASUREMENT AT THE ABLATION TABLE'S PRICE

Enumeration is hopeless at twenty guards. Three budgets, in the order a team pays for them: the ablation table itself, $n{+}1$ executions yielding $v(N)$ and every $v(N \setminus \{i\})$; plus a *typed first-blocker log*, which a short-circuiting stack keeps at no extra cost; plus gold labels sorting a logged block into a correct refusal or an over-refusal. Recapture governs a row's support: with B_i the requests guard i blocks and $U_i \subseteq B_i$ those it blocks alone, removing i changes the decision on exactly U_i, so $v(N) - v(N \setminus \{i\})$ aggregates over U_i and nothing else, and $r_i = 1 - |U_i|/|B_i|$ is the share of i's work the rest absorbs. That is §2.1's support condition in the budget's notation. On the harmful side $U_i = \emptyset$ for every guard (§4.1 in this notation).

Leave-one-in: the other half of the sandwich, for the same $n{+}1$ runs. An $n{+}1$ budget is not the ablation table: ϕ_i averages marginal contributions over every coalition, and the ablation habit reads only the one at full membership, $v(N) - v(N \setminus \{i\})$. The one at empty membership, $v(\{i\}) - v(\emptyset)$, costs the same $n{+}1$ executions, and we had already run it as the single-guard block of Table 3.

It resolves where ablation cannot: leave-one-out is evaluated where redundancy is maximal, which on a saturated harmful side annihilates it, while leave-one-in is evaluated where redundancy is nil. Single-guard harmful pass rates run from 0.0019 (G) to 0.2242 (M) against a column identically zero. Ranking guards by their leave-one-in marginals reproduces the exact Shapley credit ordering up to the adjacent swaps L1/Q, L4/Q, M/G, where leave-one-out resolves only 4 guards and ties {L, S, L4, W, L1}, the 5 largest loss-reducers by ϕ, at exactly zero. It is not unbiased — it overstates a duplicated player as badly as leave-one-out understates its credit — but it is already paid for wherever the guards were benchmarked singly, and diagnostic, a large gap between the marginals signalling recapture.

5.3 THE OBSERVABLE RECAPTURE RATE IS A LOWER BOUND

A log reveals B_i only through the order: fix the pipeline order π and let B_i^π be the requests on which i is the first blocker. Any block of i's that a guard earlier in π also catches is credited to that guard, so $U_i \subseteq B_i^\pi \subseteq B_i$ and the rate a team can compute at the second or third budget is

$$\hat{r}_i \;=\; 1 - \frac{|U_i|}{|B_i^\pi|} \;\le\; 1 - \frac{|U_i|}{|B_i|} \;=\; r_i, \tag{2}$$

with equality when no earlier guard shadows any of i's blocks; wherever $U_i = \emptyset$ the true rate is already 1 and the observable one is undefined if the log never records i as a first blocker, which holds for all 5 zeroed guards under the roster order we log (§2.1). Gold labels leave the inequality intact within the harm- and benign-side parts. The cheap tiers see a lower bound, which fixes the direction of every conclusion drawn at them: a high $\hat{r}_i$ proves r_i is high, so guard i's row measures a set the rest covers and the logged tiers can *prove* that row rests on narrow support, so a credit reading of it is unsupported, while a low $\hat{r}_i$ proves nothing — a failure of identifiability, not precision. A guard duplicating one earlier in π and a guard that blocks nothing share every observable at the first three budgets ($U = \emptyset$, leave-one-out zero, an empty logged block set), though the first has nonzero Shapley value (Shapley, 1953) and the second is a null player. Leave-one-in separates them for the price of the singleton runs.

That is L4's situation, and why the diagnostic fails where it matters: a near-totally shadowed guard is first blocker on almost nothing, so $\hat{r}_i$ has almost no denominator — under the order we log, none at all — and a flag on high recapture cannot fire because the log holds nothing to test the row with. That row is far from its Shapley value (+0.000 against -0.084), and its logged denominator moves with the pipeline order, an implementation accident. An unmodelled always-on blocker only worsens it (§6).

Validating a credit reading is not finding a subset. The certificate at this budget is one of narrow support only, and it concerns the reading of *rows*, not the choice of *subsets*. *Validating a credit reading* asks whether $v(N) - v(N \setminus \{i\})$ represents the credit one wants to attribute to guard i. Equation 2 lets the cheap tiers show a row rests on narrow support and never that the support is representative of the credit, and a zero row stays consistent with a guard holding a large share of the credit under each Shapley normalisation of §4.3. *Finding a good subset* asks only which coalition to ship, and that was cheap here: greedy backward elimination reached the exact minimiser on every

suite lattice at every λ (§4.2); search only *compares* coalitions where attribution *prices* a player, so nothing here argues for enumerating 2^n.

What a cheap budget cannot deliver is the credit *values*: on a saturated stack its one grand-coalition marginal is structurally zero, and sampling estimators (Castro et al., 2009; Covert & Lee, 2021) still exceed $n+1$.

6 Assumptions and scope

6.1 The checklist

Eight assumptions bound the game. *Purity.* A guard votes and mutates nothing, so $v(C)$ is well defined on all 512 subsets; a stage that rewrites the request makes value stop depending on membership alone. *The value function reads the blocked set alone.* Equation 1 counts prompts, never dollars or milliseconds. Under any cost or latency term a guard whose removal changes no decision acquires a strictly positive ablation score (its own cost) and "delete the zeros" stops being wrong about *that* deletion, though not about the credit, and not jointly, since the zeroed guards recapture each other. *The composition rule is any-one-blocks.* Under k-of-n on the same decisions the harmful column is identically zero only for $k \leq 2$, and by $k = 4$ no guard is zeroed and the gap has fallen from 11.30$\times$ to 2.97$\times$: both findings are properties of small k. *One guard's tier boundary is a convention.* Blocking Q on Unsafe alone leaves $H(N)$, the sole-blocker count and the zero harmful column untouched, and turns the zeroed set over, so the minimiser is no longer inside it (Appendix F). *An unsaturated harmful side, for the safety question only.* The central zero, $H(N) = 0.0000$ with no sole blockers, holds at *these operating points* only. *Public benchmarks.* The harmful pools are public and may overlap with later guards' safety-training data, so the saturation is a property of these checkpoints on these distributions, not an estimate for unseen harmful traffic.

Additivity and sign-definiteness of the per-prompt weights. Attenuation (ablation collapsing toward zero) needs only redundancy and averaging and survives any non-negative weighting; reversal needs weights that are not sign-definite, supplied here only by the mixed-sign $H + \lambda F$. *A complete player set, and removability.* Every source of refusal must be a player. An unmodelled always-on blocker (a regex blocklist, a platform filter, a system prompt) is an invisible recapturer that absorbs the removed guard's work and corrupts every recapture rate in §5.3, the lower bound included. But the lattice prices a player by what happens in its absence and S cannot be switched off, so we report both columns (§4.3), the S-fixed one being what a team can act on.

S, our own keyword matcher, is validated only by LLM adjudication, and three rubrics returning near-identical labels measure the shared adjudicator, not independent confirmation (Zheng et al., 2023; Panickssery et al., 2024).

7 Related work and conclusion

Attribution in compound LLM systems. The cooperative-game material is credited in §2.2. As systems became pipelines of models and tools (Zaharia et al., 2024), deciding which stage to credit became a routine of removing one at a time; recent work reports leave-one-out identifying bottleneck agents about as well as combinatorial attribution (Lu et al., 2026), or uses it as the cheap proxy for an NP-hard Shapley value, each removal checked by rollback (Chen et al., 2026), or uses Shapley-style attribution for workflow model allocation and for prompt optimisation, respectively (Yang et al., 2026b; Xia et al., 2026). Those works do not study refusal stacks, in which the guards left behind can recapture a removed guard's decisions. Concurrent guardrail work finds a different failure, a system-level refusal misassigned between a defence and the model when the evaluation interface changes (Zhang et al., 2026); ours holds the interface fixed. We characterise a support-collapse regime in which the harmful side loses its boundary mass while the benign side keeps its own. Guardrail evaluations, including those we do not run (Mazeika et al., 2024; Xie et al., 2024; Wang et al., 2026), test systems rather than their components.

Conclusion. An ablation table over a saturated harmful side reads a constant: its numbers are the benign half, and an all-zero removal column cannot by itself identify any guard's share of the coverage.

AI USE STATEMENT

Generative AI is both the object of study and a tool in this work, and the two uses are separate.

As object of study. 8 players in the audited stack are released guard models run at their vendors' default operating points, and the remaining player is the generator's own refusal behaviour. Their outputs are data, not authored content.

As a research tool, with required disclosure. Of the tasks ICLR lists as requiring disclosure, we used generative AI for the following. *Feedback on research methodology and experiments*: adversarial reviews, most of them generated by an AI assistant run repeatedly against this manuscript and its artifacts, are why several controls in this paper exist — the matched false-positive fit-set transfer, the k-of-n sweep, the coalition-size table, the one-per-bloc phase control, the Qwen tier-boundary control, the nine-player detector stress, the source-stratified bootstrap and the crossed one-per-bloc calibration control among them. *Implementing methods*: the analysis and table-generation code was written with an AI coding assistant. *Interpreting results*: the same assistant was used to stress-test readings of the numbers and to narrow claims that the artifacts did not support; a list of what that narrowing withdrew is in the released artifact (`ARTIFACT_AUDIT.md`; see Appendix H). *Conceptual framing and hypothesis refinement*: the framing of the harmful side as a support problem rather than an estimation problem came out of that process. No numerical result was taken from model-generated prose or kept without recomputation: every figure the paper prints is emitted from the run artifacts into a single macro file and referenced by name, and `regen_all_v2.sh` and `check_numbers_v1.py` regenerate all of them from the released records and fail on any mismatch, so a printed figure the artifacts do not support cannot survive the check. The authors chose which suggestions to run, ran them, and decided what to keep, withdraw and print. *Mathematical claims*: AI-assisted adversarial review proposed the counting corollary of Appendix A (the removal-support count) during revision; it is stated as a corollary of the sole-blocker condition, not as a theorem, and was checked by exhaustive enumeration before it was kept; that check now runs on every regeneration in the released code (`support_law_v1.py`). We used generative AI for none of the other listed tasks: no synthetic data, no proof beyond the one-sentence counting argument of that corollary, no translation, no dataset cleaning or reformatting beyond the adjudicated label corrections disclosed next (the StrongREJECT redaction is a licence step performed by our own script), and no qualitative or thematic analysis beyond that same LLM adjudication of detector labels.

LLM adjudication, as a measurement. The self-refusal detector's validation labels are LLM-adjudicated rather than human. Three differently framed rubrics were run over stratified samples in three rounds, and 20 adjudicated corrections were applied to the lattice. We do not present the resulting agreement statistics as evidence of independent confirmation: the three rubrics returned near-identical label vectors, which measures the shared adjudicator rather than independent judgement, and we say so in §6 and Appendix C. The adjudication records are released so the labels can be checked against a human standard we did not have. No other measurement in the paper depends on an LLM judgement: the coalition lattices, the Shapley values, the leave-one-out scores and the bootstrap are deterministic functions of stored decision records (Appendix F).

As a research tool, with recommended disclosure. We used an AI coding assistant while writing the analysis and paper-generation code, while drafting and revising the manuscript, and to locate related work and re-check each citation, every one of which was then verified against its primary source. We reviewed all AI-assisted work. Every number the paper prints is emitted from the run artifacts as above, so a drafting error cannot introduce a figure that the artifacts do not support; the claims, the experimental design and the decisions about what to withdraw are the authors'.

We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

No human subjects were involved and no personal data was collected; the LLM adjudication described above replaces the human annotation we could not obtain, and its limits are stated rather than papered over.

The paper reports harmful-prompt block rates for 8 named released guard models, which is information about the weaknesses of deployed safety components. We think publishing it is right on balance: all four prompt suites are already public and were built to be evaluated against (Souly et al., 2024; Röttger et al., 2024; Chao et al., 2024; Cui et al., 2025; An et al., 2024), the numbers we report are aggregate rather than a recipe, and the finding a defender needs — that a full-stack ablation of a saturated stack cannot tell them which guard supplies the screening — cannot be stated without them. The released records are per prompt, as reproducing those numbers requires: each carries each guard's decision. We redistribute prompt text under each suite's own licence, except for 92 StrongREJECT prompts its authors collected from earlier datasets, which keep their original licences, several of them unstated: for those the release carries a hash and a script that restores the text from the official file; every reported number is reproduced from the stored decision records without those texts (Appendix F).

One result invites a misreading we want to foreclose. A subset of 1 attains lower loss than the 9-guard stack we assembled, but that stack is ours, is not anyone's production system, and by its own lattice has only the empty coalition below it, 31 coalitions tied with it (§2.1); the optimum is also benchmark-dependent (§4.4). Nothing here licenses removing guards from a deployed system on the strength of an ablation table — the paper's argument is that such a table is exactly what one should not prune on.

REPRODUCIBILITY STATEMENT

The audited stack, the value function and the four prompt suites are specified in §2.1; the cluster construction behind every interval is in Appendix B with its seed. The complete 512-coalition lattice is tabulated in Appendix A, so the Shapley values and leave-one-out scores can be recomputed from the printed table alone. The detector that supplies that player is validated in Appendix C, whose figures are derived from the released adjudication records rather than transcribed. Appendix F describes the record schema and the artifact structure, and states what we can and cannot redistribute: the guard decision records reproduce every number in the main text without re-running a model, while the Korean public-health cascade of Appendix E ships its pre-computed coalition lattices and selective-prediction summaries but not its retrieval stores. Controls that did not go our way are reported alongside the ones that did (Appendices F and G).

REFERENCES

Bang An, Sicheng Zhu, Ruiyi Zhang, Michael-Andrei Panaitescu-Liess, Yuancheng Xu, and Furong Huang. Automatic pseudo-harmful prompt generation for evaluating false refusals in large language models. In *Proceedings of the Conference on Language Modeling (COLM)*, 2024. URL `https://openreview.net/forum?id=ljFgX6A8NL`.

John F. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19 (2):317–343, 1965. Print-only law review; not indexed by Crossref.

Z. W. Birnbaum. On the importance of different components in a multicomponent system. In P. R. Krishnaiah (ed.), *Multivariate Analysis II*, pp. 581–592. Academic Press, New York, 1969.

A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. Bootstrap-based improvements for inference with clustered errors. *The Review of Economics and Statistics*, 90(3):414–427, 2008. doi: 10.1162/rest.90.3.414.

Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009. doi: 10.1016/j.cor.2008.04.004.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024) Datasets and Benchmarks Track*, 2024.

Yulang Chen, Haoxuan Peng, Jinyan Liu, Zichen Wen, Dongrui Liu, and Linfeng Zhang. AgentSlimming: Towards efficient and cost-aware multi-agent systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30064–30086, 2026.

Hana Chockler and Joseph Y. Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004. doi: 10.1613/jair.1391.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104.

Ian Covert and Su-In Lee. Improving KernelSHAP: Practical Shapley value estimation using linear regression. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3457–3465. PMLR, 2021. URL https://proceedings.mlr.press/v130/covert21a.html.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021. URL https://jmlr.org/papers/v22/20-1316.html.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. OR-Bench: An over-refusal benchmark for large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 11515–11542. PMLR, 2025. URL https://proceedings.mlr.press/v267/cui25a.html.

Pradeep Dubey and Lloyd S. Shapley. Mathematical properties of the Banzhaf power index. *Mathematics of Operations Research*, 4(2):99–131, 1979. doi: 10.1287/moor.4.2.99.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010. URL https://www.jmlr.org/papers/v11/el-yaniv10a.html.

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory*, 28(4):547–565, 1999. doi: 10.1007/s001820050125.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models. arXiv preprint arXiv:2407.21783v3, 2024. URL https://arxiv.org/abs/2407.21783v3. Technical report; version v3, 23 November 2024; 560 authors.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024) Datasets and Benchmarks Track*, pp. 8093–8131, 2024. doi: 10.52202/079017-0261.

John C. Harsanyi. A simplified bargaining model for the n-person cooperative game. *International Economic Review*, 4(2):194–220, 1963. doi: 10.2307/2525487.

Daniel G. Horvitz and Donovan J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952. doi: 10.1080/01621459.1952.10483446.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama Guard: LLM-based input-output safeguard for human-AI conversations. arXiv preprint arXiv:2312.06674v1, 2023. URL https://arxiv.org/abs/2312.06674v1. Version v1, 7 December 2023.

Mingyu Lu, Yushan Huang, Chris Lin, and Su-In Lee. Agents that matter: Optimizing multi-agent LLMs via removal-based attribution. arXiv:2605.27621v1, 2026. URL https://arxiv.org/abs/2605.27621v1. arXiv preprint, version v1, 26 May 2026.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 4765–4774, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Meta. Llama Guard 3-1B model card. Model card, meta-llama/Llama-Guard-3-1B, Hugging Face, 2024. URL https://huggingface.co/meta-llama/Llama-Guard-3-1B. Repository created 20 September 2024; no accompanying paper.

Meta. Llama Guard 4 model card. Model card, meta-llama/Llama-Guard-4-12B, Hugging Face, 2025. URL https://huggingface.co/meta-llama/Llama-Guard-4-12B. Released 29 April 2025; no accompanying paper.

Guillermo Owen. Multilinear extensions of games. *Management Science*, 18(5-part-2):64–79, 1972. doi: 10.1287/mnsc.18.5.64.

Guillermo Owen. Values of games with a priori unions. In *Essays in Mathematical Economics and Game Theory*, Lecture Notes in Economics and Mathematical Systems, pp. 76–88. Springer, 1977. doi: 10.1007/978-3-642-45494-3_7.

Inkit Padhi, Manish Nagireddy, Giandomenico Cornacchia, Subhajit Chaudhury, Tejaswini Pedapati, Pierre Dognin, Keerthiram Murugesan, Erik Miehling, Martín Santillán Cooper, Kieran Fraser, Giulio Zizzo, Muhammad Zaid Hameed, Mark Purcell, Michael Desmond, Qian Pan, Inge Vejsbjerg, Elizabeth M. Daly, Michael Hind, Werner Geyer, Ambrish Rawat, Kush R. Varshney, and Prasanna Sattigeri. Granite Guardian: Comprehensive LLM safeguarding. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pp. 607–615, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-industry.49. URL https://aclanthology.org/2025.naacl-industry.49/.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7f1f0218e45f5414c79c0679633e47bc-Abstract-Conference.html.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, Mexico City, Mexico, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games, Volume II*, number 28 in Annals of Mathematics Studies, pp. 307–317. Princeton University Press, Princeton, NJ, 1953.

Lloyd S. Shapley. Cores of convex games. *International Journal of Game Theory*, 1(1):11–26, 1971. doi: 10.1007/BF01753431.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A StrongREJECT for empty jailbreaks. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024) Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e2e06adf560b0706d3b1ddfca9f29756-Abstract-Datasets_and_Benchmarks_Track.html.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The Shapley Taylor interaction index. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9259–9268. PMLR, 2020. URL https://proceedings.mlr.press/v119/sundararajan20a.html.

Xunguang Wang, Zhenlan Ji, Wenxuan Wang, Zongjie Li, Daoyuan Wu, and Shuai Wang. SoK: Evaluating jailbreak guardrails for large language models. In *IEEE Symposium on Security and Privacy (S&P)*, 2026. arXiv:2506.10597.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

Yihan Xia, Taotao Wang, Shengli Zhang, Zhangyuhua Weng, Bin Cao, and Soung Chang Liew. HiveMind: Contribution-guided online prompt optimization of LLM multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40(35), pp. 29767–29774, 2026. doi: 10.1609/aaai.v40i35.40222.

Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, Ruoxi Jia, Bo Li, Kai Li, Danqi Chen, Peter Henderson, and Prateek Mittal. SORRY-Bench: Systematically evaluating large language model safety refusal. *arXiv preprint arXiv:2406.14598*, 2024.

Boyu Yang, Jiazheng Sun, Zilong Lu, Zhi Qiu, Xin Peng, and Jun Zheng. MeClear: Cooperative game-theoretic attribution and risk-aware memory clearance for long-horizon LLM agents. *arXiv preprint arXiv:2609.09115*, 2026a.

Yingxuan Yang, Bo Huang, Siyuan Qi, Chao Feng, Haoyi Hu, Yuxuan Zhu, Jinbo Hu, Haoran Zhao, Ziyi He, Xiao Liu, ZongYu Wang, Muning Wen, Lin Qiu, Xuezhi Cao, Xunliang Cai, Yong Yu, and Weinan Zhang. Attribution-based analysis and optimization of modular agentic workflows. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 7247–7264, 2026b.

Matei Zaharia, Omar Khattab, Lingjiao Chen, Jared Quincy Davis, Heather Miller, Christopher Potts, James Zou, Michael Carbin, Jonathan Frankle, Naveen Rao, and Ali Ghodsi. The shift from models to compound AI systems. Berkeley Artificial Intelligence Research (BAIR) blog, 2024. URL https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/. Blog post, 18 February 2024.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, Olivia Sturman, and Oscar Wahltinez. ShieldGemma: Generative AI content moderation based on Gemma. arXiv preprint arXiv:2407.21772v2, 2024. URL https://arxiv.org/abs/2407.21772v2. Technical report; version v2, 4 August 2024.

Haoyu Zhang, Xiao Luo, Haowen Xu, Yi Feng, Shibo Zheng, Mohammad Zandsalimy, and Shanu Sushmita. Whose refusal is it? the unmeasured contribution of black-box multimodal guardrails. *arXiv preprint arXiv:2608.08641v1*, 2026.

Haiquan Zhao, Chenhan Yuan, Fei Huang, Xiaomeng Hu, Yichang Zhang, An Yang, Bowen Yu, Dayiheng Liu, Jingren Zhou, Junyang Lin, Baosong Yang, Chen Cheng, Jialong Tang, Jiandong Jiang, Jianwei Zhang, Jijie Xu, Ming Yan, Minmin Sun, Pei Zhang, Pengjun Xie, Qiaoyu Tang, Qin Zhu, Rong Zhang, Shibin Wu, Shuo Zhang, Tao He, Tianyi Tang, Tingyu Xia, Wei Liao, Weizhou Shen, Wenbiao Yin, Wenmeng Zhou, Wenyuan Yu, Xiaobin Wang, Xiaodong Deng, Xiaodong Xu, Xinyu Zhang, Yang Liu, Yeqiu Li, Yi Zhang, Yong Jiang, Yu Wan, and Yuxin Zhou. Qwen3Guard technical report. arXiv preprint arXiv:2510.14276v1, 2025. URL https://arxiv.org/abs/2510.14276v1. Technical report; version v1, 16 October 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023) Datasets and Benchmarks Track*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

A THE COMPLETE COALITION LATTICE AND PER-GUARD CONFUSION COUNTS

What was executed, and what was derived. The 9 guards were each executed once on every prompt of every suite. A coalition's decision is then not a separate inference run: under any-one-blocks, coalition C blocks prompt p if and only if some $i \in C$ blocks p, so the decision of every one of the 512 coalitions is a Boolean OR over columns of a stored decision matrix, one row per prompt and one column per guard. This is the sense in which all 512 coalitions were enumerated rather than simulated: no guard behaviour is modelled, imputed or interpolated, and the only assumption between the stored matrix and Table 3 is the composition rule the full stack we assembled uses. Given the matrix, the lattice is exact and re-derivable in seconds (Appendix F).

The fourth suite has a benign side only. Three of the four suites carry both pools, so each has a lattice in H and F. PHTest does not: both its tiers are benign, H is undefined on it, and the per-suite lattice our pipeline builds requires both. The setup section counts it among the suites the coalitions were run on, which was true of the guards before it was true of the lattice, so we enumerate the side that is defined. Over 1000 PHTest prompts the full stack blocks 0.8780; G alone blocks 0.8050 of them and the cheapest single guard, L, blocks 0.1230. The pathology this paper is about reappears on the benign side of this suite too: 3 guards (L, G8, L4) have a leave-one-out of exactly zero there, having no prompt they block alone.

On the main suite the harmful pool is StrongREJECT plus XSTest-unsafe (313+200 items) and the benign pool is XSTest-safe (250). For a coalition C, $H(C)$ is the fraction of the harmful pool C fails to block and $F(C)$ the fraction of the benign pool it blocks, so the empty coalition sits at $H = 1$, $F = 0$ and the loss $v = H + \lambda F$ is maximal on the harm side and zero on the cost side. All 512 coalitions were enumerated this way and all 512 are printed here, beginning at Table 4 and running in coalition-size order to the full stack in the last row of the last part. The build asserts that the number of rows printed equals the size of the lattice, so a truncated appendix cannot pass for a complete one. Each row carries the coalition's raw harmful- and benign-block counts n_H and n_F and its loss v at $\lambda = 1$. The rows the argument in §4.1 turns on — the 9 singletons, the full stack, and {W} — are therefore read out of those tables rather than asserted, and so is every other lattice quantity we quote: H and F are the two counts over the pool sizes given in the caption, and the loss gap is a ratio of two entries in a v column. The same enumeration in machine-readable form is in the released lattice file (Appendix F), from which both tables are regenerated.

Table 4: Every one of the 512 coalitions of the 9-guard stack on the main suite, ordered by coalition size and then lexically, with the raw block counts the lattice is computed from. n_H is the number of harmful prompts the coalition blocks, out of 313+200; n_F the number of benign prompts it blocks, out of 250. So $H = 1 - n_H/(313{+}200)$, $F = n_F/250$, and the last column is the loss $v = H + \lambda F$ at $\lambda = 1$: every lattice quantity the paper quotes is recomputable from these tables alone. Coalition sizes 0 to 9, continued over the next 7 tables (5 onward).

coalition	n_H	n_F	v	coalition	n_H	n_F	v
{}	0	0	1.0000	{M,L1}	489	78	0.3588
{G8}	511	77	0.3119	{M,L4}	485	81	0.3786
{G}	512	99	0.3979	{M,Q}	507	88	0.3637
{L1}	449	16	0.1888	{M,W}	504	69	0.2935
{L4}	441	17	0.2084	{Q,L1}	506	44	0.1896
{L}	468	7	0.1157	{Q,L4}	505	43	0.1876
{M}	398	68	0.4962	{Q,W}	508	35	0.1497
{Q}	504	35	0.1575	{S,G8}	512	78	0.3139
{S}	496	15	0.0931	{S,L1}	505	23	0.1076
{W}	493	3	0.0510	{S,L4}	504	26	0.1215
{G,G8}	513	121	0.4840	{S,M}	509	78	0.3198
{G,L1}	512	99	0.3979	{S,Q}	513	45	0.1800
{G,L4}	512	100	0.4019	{S,W}	511	18	0.0759
{G,L}	512	99	0.3979	{W,L1}	498	19	0.1052
{G,M}	513	131	0.5240	{G,G8,L1}	513	121	0.4840
{G,Q}	513	112	0.4480	{G,G8,L4}	513	121	0.4840
{G,S}	512	99	0.3979	{G,G8,Q}	513	130	0.5200
{G,W}	512	101	0.4059	{G,G8,W}	513	122	0.4880
{G8,L1}	512	79	0.3179	{G,L,G8}	513	121	0.4840
{G8,L4}	511	78	0.3159	{G,L,L1}	512	99	0.3979
{G8,Q}	512	91	0.3659	{G,L,L4}	512	100	0.4019
{G8,W}	512	78	0.3139	{G,L,M}	513	131	0.5240
{L,G8}	511	77	0.3119	{G,L,Q}	513	112	0.4480
{L,L1}	477	17	0.1382	{G,L,S}	512	99	0.3979
{L,L4}	479	19	0.1423	{G,L,W}	512	101	0.4059
{L,M}	493	73	0.3310	{G,L4,L1}	512	100	0.4019
{L,Q}	505	37	0.1636	{G,L4,W}	512	101	0.4059
{L,S}	506	17	0.0816	{G,M,G8}	513	139	0.5560
{L,W}	501	10	0.0634	{G,M,L1}	513	131	0.5240
{L4,L1}	467	26	0.1937	{G,M,L4}	513	131	0.5240
{L4,W}	499	19	0.1033	{G,M,Q}	513	137	0.5480
{M,G8}	512	107	0.4299	{G,M,W}	513	132	0.5280

Table 5: The complete coalition lattice, continued from Table 4 (part 2 of 8). Columns, prompt pools and the definition of v are as in Table 4.

coalition	n_H	n_F	v	coalition	n_H	n_F	v
{G,Q,L1}	513	112	0.4480	{L,S,L4}	509	26	0.1118
{G,Q,L4}	513	112	0.4480	{L,S,M}	511	80	0.3239
{G,Q,W}	513	112	0.4480	{L,S,Q}	513	45	0.1800
{G,S,G8}	513	121	0.4840	{L,S,W}	512	20	0.0819
{G,S,L1}	512	99	0.3979	{L,W,L1}	501	20	0.1034
{G,S,L4}	512	100	0.4019	{L4,W,L1}	499	28	0.1393
{G,S,M}	513	131	0.5240	{M,G8,L1}	513	108	0.4320
{G,S,Q}	513	112	0.4480	{M,G8,L4}	512	108	0.4339
{G,S,W}	512	101	0.4059	{M,G8,Q}	513	116	0.4640
{G,W,L1}	512	101	0.4059	{M,G8,W}	513	108	0.4320
{G8,L4,L1}	512	80	0.3219	{M,L4,L1}	494	85	0.3770
{G8,L4,W}	512	79	0.3179	{M,L4,W}	507	82	0.3397
{G8,Q,L1}	512	93	0.3739	{M,Q,L1}	508	93	0.3817
{G8,Q,L4}	512	91	0.3659	{M,Q,L4}	507	95	0.3917
{G8,Q,W}	512	91	0.3659	{M,Q,W}	510	88	0.3578
{G8,W,L1}	512	80	0.3219	{M,W,L1}	507	79	0.3277
{L,G8,L1}	512	79	0.3179	{Q,L4,L1}	506	50	0.2136
{L,G8,L4}	511	78	0.3159	{Q,L4,W}	508	43	0.1817
{L,G8,Q}	512	91	0.3659	{Q,W,L1}	508	44	0.1857
{L,G8,W}	512	78	0.3139	{S,G8,L1}	513	80	0.3200
{L,L4,L1}	482	27	0.1684	{S,G8,L4}	512	79	0.3179
{L,L4,W}	502	21	0.1054	{S,G8,Q}	513	92	0.3680
{L,M,G8}	512	107	0.4299	{S,G8,W}	513	79	0.3160
{L,M,L1}	498	79	0.3452	{S,L4,L1}	507	31	0.1357
{L,M,L4}	496	82	0.3611	{S,L4,W}	511	28	0.1159
{L,M,Q}	507	89	0.3677	{S,M,G8}	512	107	0.4299
{L,M,W}	509	74	0.3038	{S,M,L1}	512	83	0.3339
{L,Q,L1}	506	44	0.1896	{S,M,L4}	511	86	0.3479
{L,Q,L4}	505	44	0.1916	{S,M,Q}	513	94	0.3760
{L,Q,W}	508	37	0.1577	{S,M,W}	512	79	0.3179
{L,S,G8}	512	78	0.3139	{S,Q,L1}	513	49	0.1960
{L,S,L1}	509	23	0.0998	{S,Q,L4}	513	50	0.2000

Table 6: The complete coalition lattice, continued from Table 4 (part 3 of 8). Columns, prompt pools and the definition of v are as in Table 4.

coalition	n_H	n_F	v	coalition	n_H	n_F	v
{S,Q,W}	513	45	0.1800	{G,M,G8,Q}	513	144	0.5760
{S,W,L1}	511	26	0.1079	{G,M,G8,W}	513	140	0.5600
{G,G8,L4,L1}	513	121	0.4840	{G,M,L4,L1}	513	131	0.5240
{G,G8,L4,W}	513	122	0.4880	{G,M,L4,W}	513	132	0.5280
{G,G8,Q,L1}	513	130	0.5200	{G,M,Q,L1}	513	137	0.5480
{G,G8,Q,L4}	513	130	0.5200	{G,M,Q,L4}	513	137	0.5480
{G,G8,Q,W}	513	130	0.5200	{G,M,Q,W}	513	137	0.5480
{G,G8,W,L1}	513	122	0.4880	{G,M,W,L1}	513	132	0.5280
{G,L,G8,L1}	513	121	0.4840	{G,Q,L4,L1}	513	112	0.4480
{G,L,G8,L4}	513	121	0.4840	{G,Q,L4,W}	513	112	0.4480
{G,L,G8,Q}	513	130	0.5200	{G,Q,W,L1}	513	112	0.4480
{G,L,G8,W}	513	122	0.4880	{G,S,G8,L1}	513	121	0.4840
{G,L,L4,L1}	512	100	0.4019	{G,S,G8,L4}	513	121	0.4840
{G,L,L4,W}	512	101	0.4059	{G,S,G8,Q}	513	130	0.5200
{G,L,M,G8}	513	139	0.5560	{G,S,G8,W}	513	122	0.4880
{G,L,M,L1}	513	131	0.5240	{G,S,L4,L1}	512	100	0.4019
{G,L,M,L4}	513	131	0.5240	{G,S,L4,W}	512	101	0.4059
{G,L,M,Q}	513	137	0.5480	{G,S,M,G8}	513	139	0.5560
{G,L,M,W}	513	132	0.5280	{G,S,M,L1}	513	131	0.5240
{G,L,Q,L1}	513	112	0.4480	{G,S,M,L4}	513	131	0.5240
{G,L,Q,L4}	513	112	0.4480	{G,S,M,Q}	513	137	0.5480
{G,L,Q,W}	513	112	0.4480	{G,S,M,W}	513	132	0.5280
{G,L,S,G8}	513	121	0.4840	{G,S,Q,L1}	513	112	0.4480
{G,L,S,L1}	512	99	0.3979	{G,S,Q,L4}	513	112	0.4480
{G,L,S,L4}	512	100	0.4019	{G,S,Q,W}	513	112	0.4480
{G,L,S,M}	513	131	0.5240	{G,S,W,L1}	512	101	0.4059
{G,L,S,Q}	513	112	0.4480	{G8,L4,W,L1}	512	81	0.3259
{G,L,S,W}	512	101	0.4059	{G8,Q,L4,L1}	512	93	0.3739
{G,L,W,L1}	512	101	0.4059	{G8,Q,L4,W}	512	91	0.3659
{G,L4,W,L1}	512	101	0.4059	{G8,Q,W,L1}	512	93	0.3739
{G,M,G8,L1}	513	139	0.5560	{L,G8,L4,L1}	512	80	0.3219
{G,M,G8,L4}	513	139	0.5560	{L,G8,L4,W}	512	79	0.3179

Table 7: The complete coalition lattice, continued from Table 4 (part 4 of 8). Columns, prompt pools and the definition of v are as in Table 4.

coalition	n_H	n_F	v	coalition	n_H	n_F	v
{L,G8,Q,L1}	512	93	0.3739	{L,S,W,L1}	512	26	0.1059
{L,G8,Q,L4}	512	91	0.3659	{M,G8,L4,L1}	513	109	0.4360
{L,G8,Q,W}	512	91	0.3659	{M,G8,L4,W}	513	109	0.4360
{L,G8,W,L1}	512	80	0.3219	{M,G8,Q,L1}	513	117	0.4680
{L,L4,W,L1}	502	29	0.1374	{M,G8,Q,L4}	513	116	0.4640
{L,M,G8,L1}	513	108	0.4320	{M,G8,Q,W}	513	116	0.4640
{L,M,G8,L4}	512	108	0.4339	{M,G8,W,L1}	513	109	0.4360
{L,M,G8,Q}	513	116	0.4640	{M,L4,W,L1}	507	86	0.3557
{L,M,G8,W}	513	108	0.4320	{M,Q,L4,L1}	508	98	0.4017
{L,M,L4,L1}	499	86	0.3713	{M,Q,L4,W}	510	95	0.3858
{L,M,L4,W}	509	83	0.3398	{M,Q,W,L1}	510	93	0.3778
{L,M,Q,L1}	508	93	0.3817	{Q,L4,W,L1}	508	50	0.2097
{L,M,Q,L4}	507	95	0.3917	{S,G8,L4,L1}	513	81	0.3240
{L,M,Q,W}	510	89	0.3618	{S,G8,L4,W}	513	80	0.3200
{L,M,W,L1}	509	80	0.3278	{S,G8,Q,L1}	513	94	0.3760
{L,Q,L4,L1}	506	50	0.2136	{S,G8,Q,L4}	513	92	0.3680
{L,Q,L4,W}	508	44	0.1857	{S,G8,Q,W}	513	92	0.3680
{L,Q,W,L1}	508	44	0.1857	{S,G8,W,L1}	513	81	0.3240
{L,S,G8,L1}	513	80	0.3200	{S,L4,W,L1}	511	33	0.1359
{L,S,G8,L4}	512	79	0.3179	{S,M,G8,L1}	513	108	0.4320
{L,S,G8,Q}	513	92	0.3680	{S,M,G8,L4}	512	108	0.4339
{L,S,G8,W}	513	79	0.3160	{S,M,G8,Q}	513	116	0.4640
{L,S,L4,L1}	510	31	0.1298	{S,M,G8,W}	513	108	0.4320
{L,S,L4,W}	512	28	0.1139	{S,M,L4,L1}	512	88	0.3539
{L,S,M,G8}	512	107	0.4299	{S,M,L4,W}	512	87	0.3499
{L,S,M,L1}	513	83	0.3320	{S,M,Q,L1}	513	96	0.3840
{L,S,M,L4}	512	86	0.3459	{S,M,Q,L4}	513	98	0.3920
{L,S,M,Q}	513	94	0.3760	{S,M,Q,W}	513	94	0.3760
{L,S,M,W}	513	81	0.3240	{S,M,W,L1}	512	84	0.3379
{L,S,Q,L1}	513	49	0.1960	{S,Q,L4,L1}	513	53	0.2120
{L,S,Q,L4}	513	50	0.2000	{S,Q,L4,W}	513	50	0.2000
{L,S,Q,W}	513	45	0.1800	{S,Q,W,L1}	513	49	0.1960

Table 8: The complete coalition lattice, continued from Table 4 (part 5 of 8). Columns, prompt pools and the definition of v are as in Table 4.

coalition	n_H	n_F	v	coalition	n_H	n_F	v
{G,G8,L4,W,L1}	513	122	0.4880	{G,L,S,M,L4}	513	131	0.5240
{G,G8,Q,L4,L1}	513	130	0.5200	{G,L,S,M,Q}	513	137	0.5480
{G,G8,Q,L4,W}	513	130	0.5200	{G,L,S,M,W}	513	132	0.5280
{G,G8,Q,W,L1}	513	130	0.5200	{G,L,S,Q,L1}	513	112	0.4480
{G,L,G8,L4,L1}	513	121	0.4840	{G,L,S,Q,L4}	513	112	0.4480
{G,L,G8,L4,W}	513	122	0.4880	{G,L,S,Q,W}	513	112	0.4480
{G,L,G8,Q,L1}	513	130	0.5200	{G,L,S,W,L1}	512	101	0.4059
{G,L,G8,Q,L4}	513	130	0.5200	{G,M,G8,L4,L1}	513	139	0.5560
{G,L,G8,Q,W}	513	130	0.5200	{G,M,G8,L4,W}	513	140	0.5600
{G,L,G8,W,L1}	513	122	0.4880	{G,M,G8,Q,L1}	513	144	0.5760
{G,L,L4,W,L1}	512	101	0.4059	{G,M,G8,Q,L4}	513	144	0.5760
{G,L,M,G8,L1}	513	139	0.5560	{G,M,G8,Q,W}	513	144	0.5760
{G,L,M,G8,L4}	513	139	0.5560	{G,M,G8,W,L1}	513	140	0.5600
{G,L,M,G8,Q}	513	144	0.5760	{G,M,L4,W,L1}	513	132	0.5280
{G,L,M,G8,W}	513	140	0.5600	{G,M,Q,L4,L1}	513	137	0.5480
{G,L,M,L4,L1}	513	131	0.5240	{G,M,Q,L4,W}	513	137	0.5480
{G,L,M,L4,W}	513	132	0.5280	{G,M,Q,W,L1}	513	137	0.5480
{G,L,M,Q,L1}	513	137	0.5480	{G,Q,L4,W,L1}	513	112	0.4480
{G,L,M,Q,L4}	513	137	0.5480	{G,S,G8,L4,L1}	513	121	0.4840
{G,L,M,Q,W}	513	137	0.5480	{G,S,G8,L4,W}	513	122	0.4880
{G,L,M,W,L1}	513	132	0.5280	{G,S,G8,Q,L1}	513	130	0.5200
{G,L,Q,L4,L1}	513	112	0.4480	{G,S,G8,Q,L4}	513	130	0.5200
{G,L,Q,L4,W}	513	112	0.4480	{G,S,G8,Q,W}	513	130	0.5200
{G,L,Q,W,L1}	513	112	0.4480	{G,S,G8,W,L1}	513	122	0.4880
{G,L,S,G8,L1}	513	121	0.4840	{G,S,L4,W,L1}	512	101	0.4059
{G,L,S,G8,L4}	513	121	0.4840	{G,S,M,G8,L1}	513	139	0.5560
{G,L,S,G8,Q}	513	130	0.5200	{G,S,M,G8,L4}	513	139	0.5560
{G,L,S,G8,W}	513	122	0.4880	{G,S,M,G8,Q}	513	144	0.5760
{G,L,S,L4,L1}	512	100	0.4019	{G,S,M,G8,W}	513	140	0.5600
{G,L,S,L4,W}	512	101	0.4059	{G,S,M,L4,L1}	513	131	0.5240
{G,L,S,M,G8}	513	139	0.5560	{G,S,M,L4,W}	513	132	0.5280
{G,L,S,M,L1}	513	131	0.5240	{G,S,M,Q,L1}	513	137	0.5480

Table 9: The complete coalition lattice, continued from Table 4 (part 6 of 8). Columns, prompt pools and the definition of v are as in Table 4.

coalition	n_H	n_F	v	coalition	n_H	n_F	v
{G,S,M,Q,L4}	513	137	0.5480	{L,S,M,G8,W}	513	108	0.4320
{G,S,M,Q,W}	513	137	0.5480	{L,S,M,L4,L1}	513	88	0.3520
{G,S,M,W,L1}	513	132	0.5280	{L,S,M,L4,W}	513	87	0.3480
{G,S,Q,L4,L1}	513	112	0.4480	{L,S,M,Q,L1}	513	96	0.3840
{G,S,Q,L4,W}	513	112	0.4480	{L,S,M,Q,L4}	513	98	0.3920
{G,S,Q,W,L1}	513	112	0.4480	{L,S,M,Q,W}	513	94	0.3760
{G8,Q,L4,W,L1}	512	93	0.3739	{L,S,M,W,L1}	513	84	0.3360
{L,G8,L4,W,L1}	512	81	0.3259	{L,S,Q,L4,L1}	513	53	0.2120
{L,G8,Q,L4,L1}	512	93	0.3739	{L,S,Q,L4,W}	513	50	0.2000
{L,G8,Q,L4,W}	512	91	0.3659	{L,S,Q,W,L1}	513	49	0.1960
{L,G8,Q,W,L1}	512	93	0.3739	{M,G8,L4,W,L1}	513	110	0.4400
{L,M,G8,L4,L1}	513	109	0.4360	{M,G8,Q,L4,L1}	513	117	0.4680
{L,M,G8,L4,W}	513	109	0.4360	{M,G8,Q,L4,W}	513	116	0.4640
{L,M,G8,Q,L1}	513	117	0.4680	{M,G8,Q,W,L1}	513	117	0.4680
{L,M,G8,Q,L4}	513	116	0.4640	{M,Q,L4,W,L1}	510	98	0.3978
{L,M,G8,Q,W}	513	116	0.4640	{S,G8,L4,W,L1}	513	82	0.3280
{L,M,G8,W,L1}	513	109	0.4360	{S,G8,Q,L4,L1}	513	94	0.3760
{L,M,L4,W,L1}	509	87	0.3558	{S,G8,Q,L4,W}	513	92	0.3680
{L,M,Q,L4,L1}	508	98	0.4017	{S,G8,Q,W,L1}	513	94	0.3760
{L,M,Q,L4,W}	510	95	0.3858	{S,M,G8,L4,L1}	513	109	0.4360
{L,M,Q,W,L1}	510	93	0.3778	{S,M,G8,L4,W}	513	109	0.4360
{L,Q,L4,W,L1}	508	50	0.2097	{S,M,G8,Q,L1}	513	117	0.4680
{L,S,G8,L4,L1}	513	81	0.3240	{S,M,G8,Q,L4}	513	116	0.4640
{L,S,G8,L4,W}	513	80	0.3200	{S,M,G8,Q,W}	513	116	0.4640
{L,S,G8,Q,L1}	513	94	0.3760	{S,M,G8,W,L1}	513	109	0.4360
{L,S,G8,Q,L4}	513	92	0.3680	{S,M,L4,W,L1}	512	89	0.3579
{L,S,G8,Q,W}	513	92	0.3680	{S,M,Q,L4,L1}	513	99	0.3960
{L,S,G8,W,L1}	513	81	0.3240	{S,M,Q,L4,W}	513	98	0.3920
{L,S,L4,W,L1}	512	33	0.1339	{S,M,Q,W,L1}	513	96	0.3840
{L,S,M,G8,L1}	513	108	0.4320	{S,Q,L4,W,L1}	513	53	0.2120
{L,S,M,G8,L4}	512	108	0.4339	{G,G8,Q,L4,W,L1}	513	130	0.5200
{L,S,M,G8,Q}	513	116	0.4640	{G,L,G8,L4,W,L1}	513	122	0.4880

Table 10: The complete coalition lattice, continued from Table 4 (part 7 of 8). Columns, prompt pools and the definition of v are as in Table 4.

coalition	n_H	n_F	v	coalition	n_H	n_F	v
{G,L,G8,Q,L4,L1}	513	130	0.5200	{G,L,S,Q,L4,W}	513	112	0.4480
{G,L,G8,Q,L4,W}	513	130	0.5200	{G,L,S,Q,W,L1}	513	112	0.4480
{G,L,G8,Q,W,L1}	513	130	0.5200	{G,M,G8,L4,W,L1}	513	140	0.5600
{G,L,M,G8,L4,L1}	513	139	0.5560	{G,M,G8,Q,L4,L1}	513	144	0.5760
{G,L,M,G8,L4,W}	513	140	0.5600	{G,M,G8,Q,L4,W}	513	144	0.5760
{G,L,M,G8,Q,L1}	513	144	0.5760	{G,M,G8,Q,W,L1}	513	144	0.5760
{G,L,M,G8,Q,L4}	513	144	0.5760	{G,M,Q,L4,W,L1}	513	137	0.5480
{G,L,M,G8,Q,W}	513	144	0.5760	{G,S,G8,L4,W,L1}	513	122	0.4880
{G,L,M,G8,W,L1}	513	140	0.5600	{G,S,G8,Q,L4,L1}	513	130	0.5200
{G,L,M,L4,W,L1}	513	132	0.5280	{G,S,G8,Q,L4,W}	513	130	0.5200
{G,L,M,Q,L4,L1}	513	137	0.5480	{G,S,G8,Q,W,L1}	513	130	0.5200
{G,L,M,Q,L4,W}	513	137	0.5480	{G,S,M,G8,L4,L1}	513	139	0.5560
{G,L,M,Q,W,L1}	513	137	0.5480	{G,S,M,G8,L4,W}	513	140	0.5600
{G,L,Q,L4,W,L1}	513	112	0.4480	{G,S,M,G8,Q,L1}	513	144	0.5760
{G,L,S,G8,L4,L1}	513	121	0.4840	{G,S,M,G8,Q,L4}	513	144	0.5760
{G,L,S,G8,L4,W}	513	122	0.4880	{G,S,M,G8,Q,W}	513	144	0.5760
{G,L,S,G8,Q,L1}	513	130	0.5200	{G,S,M,G8,W,L1}	513	140	0.5600
{G,L,S,G8,Q,L4}	513	130	0.5200	{G,S,M,L4,W,L1}	513	132	0.5280
{G,L,S,G8,Q,W}	513	130	0.5200	{G,S,M,Q,L4,L1}	513	137	0.5480
{G,L,S,G8,W,L1}	513	122	0.4880	{G,S,M,Q,L4,W}	513	137	0.5480
{G,L,S,L4,W,L1}	512	101	0.4059	{G,S,M,Q,W,L1}	513	137	0.5480
{G,L,S,M,G8,L1}	513	139	0.5560	{G,S,Q,L4,W,L1}	513	112	0.4480
{G,L,S,M,G8,L4}	513	139	0.5560	{L,G8,Q,L4,W,L1}	512	93	0.3739
{G,L,S,M,G8,Q}	513	144	0.5760	{L,M,G8,L4,W,L1}	513	110	0.4400
{G,L,S,M,G8,W}	513	140	0.5600	{L,M,G8,Q,L4,L1}	513	117	0.4680
{G,L,S,M,L4,L1}	513	131	0.5240	{L,M,G8,Q,L4,W}	513	116	0.4640
{G,L,S,M,L4,W}	513	132	0.5280	{L,M,G8,Q,W,L1}	513	117	0.4680
{G,L,S,M,Q,L1}	513	137	0.5480	{L,M,Q,L4,W,L1}	510	98	0.3978
{G,L,S,M,Q,L4}	513	137	0.5480	{L,S,G8,L4,W,L1}	513	82	0.3280
{G,L,S,M,Q,W}	513	137	0.5480	{L,S,G8,Q,L4,L1}	513	94	0.3760
{G,L,S,M,W,L1}	513	132	0.5280	{L,S,G8,Q,L4,W}	513	92	0.3680
{G,L,S,Q,L4,L1}	513	112	0.4480	{L,S,G8,Q,W,L1}	513	94	0.3760

Table 11: The complete coalition lattice, continued from Table 4 (part 8 of 8), ending in the full stack in the final row. Columns, prompt pools and the definition of v are as in Table 4.

coalition	n_H	n_F	v	coalition	n_H	n_F	v
{L,S,M,G8,L4,L1}	513	109	0.4360	{G,L,S,M,G8,Q,W}	513	144	0.5760
{L,S,M,G8,L4,W}	513	109	0.4360	{G,L,S,M,G8,W,L1}	513	140	0.5600
{L,S,M,G8,Q,L1}	513	117	0.4680	{G,L,S,M,L4,W,L1}	513	132	0.5280
{L,S,M,G8,Q,L4}	513	116	0.4640	{G,L,S,M,Q,L4,L1}	513	137	0.5480
{L,S,M,G8,Q,W}	513	116	0.4640	{G,L,S,M,Q,L4,W}	513	137	0.5480
{L,S,M,G8,W,L1}	513	109	0.4360	{G,L,S,M,Q,W,L1}	513	137	0.5480
{L,S,M,L4,W,L1}	513	89	0.3560	{G,L,S,Q,L4,W,L1}	513	112	0.4480
{L,S,M,Q,L4,L1}	513	99	0.3960	{G,M,G8,Q,L4,W,L1}	513	144	0.5760
{L,S,M,Q,L4,W}	513	98	0.3920	{G,S,G8,Q,L4,W,L1}	513	130	0.5200
{L,S,M,Q,W,L1}	513	96	0.3840	{G,S,M,G8,L4,W,L1}	513	140	0.5600
{L,S,Q,L4,W,L1}	513	53	0.2120	{G,S,M,G8,Q,L4,L1}	513	144	0.5760
{M,G8,Q,L4,W,L1}	513	117	0.4680	{G,S,M,G8,Q,L4,W}	513	144	0.5760
{S,G8,Q,L4,W,L1}	513	94	0.3760	{G,S,M,G8,Q,W,L1}	513	144	0.5760
{S,M,G8,L4,W,L1}	513	110	0.4400	{G,S,M,Q,L4,W,L1}	513	137	0.5480
{S,M,G8,Q,L4,L1}	513	117	0.4680	{L,M,G8,Q,L4,W,L1}	513	117	0.4680
{S,M,G8,Q,L4,W}	513	116	0.4640	{L,S,G8,Q,L4,W,L1}	513	94	0.3760
{S,M,G8,Q,W,L1}	513	117	0.4680	{L,S,M,G8,L4,W,L1}	513	110	0.4400
{S,M,Q,L4,W,L1}	513	99	0.3960	{L,S,M,G8,Q,L4,L1}	513	117	0.4680
{G,L,G8,Q,L4,W,L1}	513	130	0.5200	{L,S,M,G8,Q,L4,W}	513	116	0.4640
{G,L,M,G8,L4,W,L1}	513	140	0.5600	{L,S,M,G8,Q,W,L1}	513	117	0.4680
{G,L,M,G8,Q,L4,L1}	513	144	0.5760	{L,S,M,Q,L4,W,L1}	513	99	0.3960
{G,L,M,G8,Q,L4,W}	513	144	0.5760	{S,M,G8,Q,L4,W,L1}	513	117	0.4680
{G,L,M,G8,Q,W,L1}	513	144	0.5760	{G,L,M,G8,Q,L4,W,L1}	513	144	0.5760
{G,L,M,Q,L4,W,L1}	513	137	0.5480	{G,L,S,G8,Q,L4,W,L1}	513	130	0.5200
{G,L,S,G8,L4,W,L1}	513	122	0.4880	{G,L,S,M,G8,L4,W,L1}	513	140	0.5600
{G,L,S,G8,Q,L4,L1}	513	130	0.5200	{G,L,S,M,G8,Q,L4,L1}	513	144	0.5760
{G,L,S,G8,Q,L4,W}	513	130	0.5200	{G,L,S,M,G8,Q,L4,W}	513	144	0.5760
{G,L,S,G8,Q,W,L1}	513	130	0.5200	{G,L,S,M,G8,Q,W,L1}	513	144	0.5760
{G,L,S,M,G8,L4,L1}	513	139	0.5560	{G,L,S,M,Q,L4,W,L1}	513	137	0.5480
{G,L,S,M,G8,L4,W}	513	140	0.5600	{G,S,M,G8,Q,L4,W,L1}	513	144	0.5760
{G,L,S,M,G8,Q,L1}	513	144	0.5760	{L,S,M,G8,Q,L4,W,L1}	513	117	0.4680
{G,L,S,M,G8,Q,L4}	513	144	0.5760	{G,L,S,M,G8,Q,L4,W,L1}	513	144	0.5760

Table 12: The lattice by coalition *size*, aggregated over all $\binom{9}{m}$ coalitions of that size on the main suite. The harmful side's removal support starts collapsing at $m = 2$: by $m = 4$, 53.6% of (coalition, member) pairs already read exactly +0.000 there. *best F at H=0* is the lowest benign block rate at that size that passes no harmful prompt.

m	coalitions	H med.	F med.	H=0	best F at H=0	≥ 2 blockers	$\mathrm{LOO}_i^H = 0$
1	9	0.0390	0.0680	0	—	0.0%	0.0%
2	36	0.0088	0.3100	4	0.1800	88.3%	9.7%
3	84	0.0019	0.3600	29	0.1800	96.5%	32.1%
4	126	0.0000	0.4040	79	0.1800	98.5%	53.6%
5	126	0.0000	0.4640	107	0.1960	99.3%	73.0%
6	84	0.0000	0.5200	81	0.2120	99.7%	87.7%
7	36	0.0000	0.5480	36	0.3760	99.9%	96.4%
8	9	0.0000	0.5760	9	0.4680	100.0%	100.0%
9	1	0.0000	0.5760	1	0.5760	100.0%	100.0%

The same lattice read by coalition size. Printing 512 rows answers what each coalition does and not what *size* does, which is the question behind the objection that a 9-way union is a stress test rather than a stack. Table 12 gives the same aggregation in full. The saturation is not an artefact of the maximum: at $m = 3$ already 32.1% of (coalition, member) pairs have a harmful leave-one-out of exactly zero. Nor does it rest on S, the one player we built rather than downloaded: across the 9 coalitions that drop a single guard, S among them, 100.0% of the remaining members still have a harmful leave-one-out of exactly zero. Nor does the cost side wait for the full stack: the lowest benign block rate that passes no harmful prompt, 0.1800, is available at $m = 2$, and the remaining $9-2$ guards multiply it by 3.20 without buying a harmful block.

The size phase, counted. The support formula is a counting corollary of the sole-blocker condition (§2.1), not a novel theoretical contribution; the measured blocker-multiplicity distribution that places released guard stacks in this regime is the result. A prompt blocked by b of the n sources gives an ordered (coalition, member) pair of size m removal support only when the member blocks it and the other $m - 1$ do not: $b\binom{n-b}{m-1}$ of the $m\binom{n}{m}$ pairs, and none once $m > n - b + 1$. Under the corrected column the 513 harmful prompts of the main suite number 331, 108, 28, 19, 16, 6, 5 at $b = 9$ down to $b = 3$, so 85.6% of them give no size-3 pair any support and 91.0% no size-4 pair. The aggregate in Table 12 is lower than those shares because a pair reads zero only when *every* prompt gives it none (`support_law_v1.py`).

Table 13: Per-guard block rate on each of the nine prompt groups of the four suites; n is the group size and a rate times n recovers the count. On a harmful group a block is a true positive and a pass a false negative; on a benign group a block is a false positive. G blocks more benign prompts than any other guard in every one of the benign groups, and the floor is held by just 3 of them (L, L4, W), so the over-refusal the leave-one-out scores in Table 3 are made of is concentrated in single guards. Rates are recomputed from the stored decision records; the JBB columns reproduce the paired-control run exactly.

Group	kind	n	G	L	S	M	G8	Q	L4	W	L1
strongreject	harmful	313	1.0000	0.9712	0.9840	0.7476	1.0000	0.9968	0.9105	0.9904	0.9553
xstest-safe	benign	250	0.3960	0.0280	0.0600	0.2720	0.3080	0.1400	0.0680	0.0120	0.0640
xstest-unsafe	harmful	200	0.9950	0.8200	0.9400	0.8200	0.9900	0.9600	0.7800	0.9150	0.7500
jbb-benign	benign	100	0.8800	0.2300	0.2900	0.2700	0.6300	0.6300	0.1400	0.4500	0.2200
jbb-harmful	harmful	100	1.0000	0.9800	0.9800	0.4900	1.0000	1.0000	0.8700	0.9900	0.9400
orbench-benign	benign	1319	0.9121	0.1865	0.4390	0.2494	0.8302	0.8196	0.1054	0.7392	0.3700
orbench-toxic	harmful	655	0.9939	0.8305	0.8260	0.5160	0.9832	0.9985	0.7237	0.9924	0.8290
phtest-controversial	benign	500	0.9060	0.1920	0.4480	0.2000	0.7380	0.6460	0.1920	0.5780	0.3640
phtest-harmless	benign	500	0.7040	0.0540	0.2280	0.1420	0.3680	0.2320	0.0720	0.2060	0.2020

Per-guard block rates. Table 13 gives, for each of the 9 guards, the fraction of items it blocks on each of the nine prompt groups of the four suites, with the group size n beside it, so a rate times n recovers the count and no separate count table is needed. On a harmful group a block is a true positive and a pass a false negative; on a benign group a block is a false positive and a pass a true negative. Two features of the table are worth naming in advance. First, in every suite that has groups of both kinds, the across-guard spread of block rates is narrower on each harmful group than on any benign group of that suite, and the harmful groups are also the ones the guards block near-unanimously (Table 21); so little column-wise variation is left on the harm side that an ablation has almost nothing to read there — this is exactly the redundancy the paper is about, and it is what makes the harm side of every leave-one-out difference vanish. Second, on the benign groups the spread is wide and ordered the same way everywhere: G blocks more benign prompts than any other guard in every one of the five benign groups, and the floor is held by just 3 of them (L, L4, W). That spread is what the differences among the leave-one-out scores in Table 3 are made of.

B SHAPLEY VALUES, LEAVE-ONE-OUT, MULTIPLICITY, AND THE BOOTSTRAP DESIGN

Estimands. For player i and the 9-player game v on N, the Shapley value is computed by exact enumeration,

$$\phi_i \;=\; \sum_{C \subseteq N \setminus \{i\}} \frac{|C|!\,(\,|N|-|C|-1)!}{|N|!}\,\big(v(C\cup\{i\})-v(C)\big),$$

over all 512 coalitions, with no sampling and no approximation (Shapley, 1953); leave-one-out is $\mathrm{LOO}_i = v(N) - v(N\setminus\{i\})$, the single term of that sum with $C = N\setminus\{i\}$, weighted by one instead of by $1/|N|$. The two are therefore on the same scale and the same sign convention: v is a loss, so negative means the player helps. The efficiency identity $\sum_i \phi_i = v(N) - v(\emptyset)$ holds to machine precision on every lattice we report, and it is the only internal check available that the enumeration is complete; we compute the residual and it is zero at every λ.

The curvature test, and why the classical dichotomy is not available here. §2.2 credits the classical result that curvature fixes the ordering between a player's marginal contribution to the grand coalition and its Shapley value (Shapley, 1971), and then declines to use it, on the ground that our game is neither convex nor concave. That is a measurement, and this is it. Convexity requires the marginal of a player to be non-decreasing along a nesting and concavity requires it to be non-increasing, so we enumerate every nested triple the definitions range over — every i and every pair $C \subsetneq D \subseteq N\setminus\{i\}$, 56745 of them on the 9-guard lattice at $\lambda = 1$ — and count the ones where i's marginal moves the wrong way for each. It moves *up* on 9386 of them, contradicting concavity, and *down* on 31105, contradicting convexity. Both counts are large, which is what a loss built as a difference of two submodular coverage functions — coverage of the harmful pool against coverage

Table 14: Exact Shapley value against leave-one-out for every guard, at the equal-weighting operating point. Intervals are percentile cluster bootstrap ($B = 6000$, 513 clusters). Ablation reports exactly zero for 5 guards ({L, S, L4, W, L1}) whose Shapley intervals exclude zero on the helpful side. The same two quantities across the whole λ grid are in Table 15.

Guard	Model	ϕ_i [95% CI]	LOO_i [95% CI]
G	granite-guardian-3.0-2b	0.084 [0.046, 0.126]	0.108 [0.071, 0.148]
L	Llama-Guard-3-8B	-0.102 [-0.106, -0.098]	0.000 [0.000, 0.000]
S	Llama-3.1-8B self-refusal	-0.105 [-0.112, -0.097]	0.000 [0.000, 0.000]
M	ShieldGemma-2B	0.046 [0.015, 0.082]	0.056 [0.030, 0.086]
G8	granite-guardian-3.0-8b	0.001 [-0.027, 0.031]	0.028 [0.008, 0.053]
Q	Qwen3Guard-Gen-4B	-0.065 [-0.084, -0.043]	0.016 [0.004, 0.033]
L4	Llama-Guard-4-12B	-0.084 [-0.091, -0.076]	0.000 [0.000, 0.000]
W	WildGuard-7B	-0.112 [-0.117, -0.106]	0.000 [0.000, 0.000]
L1	Llama-Guard-3-1B	-0.087 [-0.094, -0.080]	0.000 [0.000, 0.000]

of the benign pool — should give. The ordering $\phi_i \leq \mathrm{LOO}_i$ that does hold for all 9 guards at λ=1 is therefore a fact about this game, and we report it as one; no curvature argument delivers it.

The two attribution tables. Table 14 gives ϕ_i and LOO_i for all 9 guards at the equal-weighting operating point $\lambda = 1$, with cluster-bootstrap percentile intervals: it is the ϕ_i and LOO_i columns of Table 3 of the main text, repeated beside the by-λ grid. Table 15 then gives the same two quantities, without intervals, at every λ the paper reports, and adds beneath them the three lattice-level quantities §4.2 argues from — the set leave-one-out sends to exactly zero, the share of $\sum_i |\phi_i|$ that set carries, and the loss-minimising coalition. The ratio printed there is one normalisation of six, and none of them is primary: mixed-sign values leave no canonical denominator for a share of credit. Table 20 prints all six side by side and §4.3 draws the conclusion. Read with that caveat, the by-λ grid is the one table in this appendix a reader checking the attribution claims of §4.3 should read, because all three quantities and the per-guard disagreement that produces them are visible in it at once.

Three things are legible from that grid directly. The zeroed set is {L, S, L4, W, L1} at every λ in it, so the identity of the guards ablation cannot see is not an artefact of the equal-weighting point. The share of the Shapley mass those guards carry, under the $\sum_i |\phi_i|$ convention the grid prints never falls below 41.8% across the grid and is largest at $\lambda = 0.5$ (72.5%), only 1.0 points above the runner-up — close enough that we do not read the location of that maximum as meaningful; at $\lambda = 1$ it is 71.5% [65.0, 76.5]; under the efficiency normalisation the same set holds 115.7% at $\lambda = 1$ (Table 20). And at every λ in the grid the loss-minimising coalition is a subset of the zeroed set — {W} from 0.585 upward, that window being unbounded above, and {S,W} immediately below it — which is the lattice form of the decision §4.2 argues a team gets wrong. Note what it is not: it is a statement about *cost* at the top of the lattice, where the harmful side is saturated (Table 16) and H is constant under any one removal. Lower down it is not: {W} itself passes 0.0390. It is also conditional on the vendor-default operating points, and Appendix G reports the control that breaks it: matched to a common benign false-positive rate the guards produce a lattice whose minimiser is no longer drawn from the zeroed set. The bootstrap agrees at the top of the grid as well as at the equal-weighting point reported in §4.2: at $\lambda = 2$ the resampled minimiser is again some subset of {L, S, L4, W, L1} in 100.0% of resamples.

5 intervals in Table 14 are degenerate at zero, and the reason is structural rather than numerical. LOO_i moves only on prompts where i is the *sole* blocker: removing i from the full stack changes the system decision on exactly those prompts. {L, S, L4, W, L1} are never the sole blocker of any prompt in the main suite, on either side of the harmful/benign split, so no resample of the clusters can produce a nonzero difference and the bootstrap distribution is a point mass. That is not a narrow interval, and it should not be read as a precise measurement of a small effect. It is the absence of any resolvable quantity.

The harmful side is saturated, which is what the degenerate intervals are made of. Table 16 states the degeneracy directly, one row per enumerated lattice, at the operating points the 8 guards ship with. On all three the full stack passes no harmful prompt at all, $H(N) = 0.0000$, and no harmful prompt anywhere is blocked by a lone guard: the fewest guards blocking any single harmful item is 3 on the main suite, 5 on JailbreakBench and 2 on OR-Bench, against sole-blocked

Table 15: The same two quantities as Table 14 at every λ the paper reports, on the same 512-coalition lattice. Below the guards: the set leave-one-out sends to exactly zero, the share of $\sum_i |\phi_i|$ that set carries, and the loss-minimising coalition. The zeroed set is the same at every λ, its share of the credit never falls below 41.8% and is largest at λ =0.5 (72.5%, only 1.0 points above the runner-up, so we do not read the location of that maximum as meaningful), and at every λ the minimiser is a subset of it: the loss-minimising coalition is drawn entirely from the guards a one-shot zero-as-dead-weight rule would delete.

	$\lambda = 0.25$		$\lambda = 0.5$		$\lambda = 1.0$		$\lambda = 1.5$		$\lambda = 2.0$	
Guard	ϕ_i	LOO_i	ϕ_i	LOO_i	ϕ_i	LOO_i	ϕ_i	LOO_i	ϕ_i	LOO_i
G	-0.072	+0.027	-0.020	+0.054	+0.084	+0.108	+0.188	+0.162	+0.292	+0.216
L	-0.106	+0.000	-0.105	+0.000	-0.102	+0.000	-0.100	+0.000	-0.098	+0.000
S	-0.115	+0.000	-0.112	+0.000	-0.105	+0.000	-0.099	+0.000	-0.092	+0.000
M	-0.057	+0.014	-0.023	+0.028	+0.046	+0.056	+0.115	+0.084	+0.184	+0.112
G8	-0.092	+0.007	-0.061	+0.014	+0.001	+0.028	+0.063	+0.042	+0.124	+0.056
Q	-0.106	+0.004	-0.092	+0.008	-0.065	+0.016	-0.037	+0.024	-0.010	+0.032
L4	-0.095	+0.000	-0.091	+0.000	-0.084	+0.000	-0.076	+0.000	-0.068	+0.000
W	-0.115	+0.000	-0.114	+0.000	-0.112	+0.000	-0.110	+0.000	-0.109	+0.000
L1	-0.098	+0.000	-0.094	+0.000	-0.087	+0.000	-0.080	+0.000	-0.073	+0.000
zeroed by LOO	L, S, L4, W, L1		L, S, L4, W, L1		L, S, L4, W, L1		L, S, L4, W, L1		L, S, L4, W, L1	
share of $\sum_i \lvert\phi_i\rvert$	61.8%		72.5%		71.5%		53.6%		41.8%	
loss minimiser	{S,W}		{S,W}		{W}		{W}		{W}	

Table 16: The harmful side is saturated on every lattice we enumerate at the vendor default operating points, which is why the ablation column reads nothing there. $H(N)$ is the fraction of the harmful pool the full stack passes; *passed by all* counts the prompts no guard blocks; *min blockers* is the smallest number of guards blocking any single harmful prompt in the pool, and *sole-blocked* counts the prompts exactly one guard blocks. That last column stands at zero on every suite — at these guards' shipped operating points no harmful prompt in this study has a sole blocker — so $H(N) = H(N \setminus \{i\})$ exactly for every guard i and the harm half of every leave-one-out difference at the vendor defaults is identically zero. PHTest ships no harmful tier and so has no row here. This is a property of these runs at these operating points, not an estimate, and not a property of the any-one-blocks rule: re-thresholded to a matched benign FPR the same 9 guards do leave sole blockers (Table 1).

harmful pool	n	$H(N)$	passed by all 9	min blockers	sole-blocked
main (SR + XSTest-unsafe)	513	0.0000	0	3	0
JBB harmful	100	0.0000	0	5	0
OR-Bench toxic	655	0.0000	0	2	0

counts of 0, 0 and 0 respectively. Two consequences follow arithmetically rather than statistically. $H(N) = H(N \setminus \{i\})$ for every guard i, so the harm half of every leave-one-out difference at the vendor defaults is identically zero and each LOO_i is λ times a difference in benign block rates. And no resampling scheme can restore what is missing: the bootstrap intervals of Table 14 are degenerate for {L, S, L4, W, L1} because the quantity being resampled is a constant, not because B is too small. We report the saturation as the paper's finding rather than as a resolution problem, because on this data it is the answer to the coverage question: harmful-prompt screening is covered by all 9 guards redundantly, and there is nothing left over for ablation to divide. The scope of that finding is the operating points these 8 guards ship with, and not the any-one-blocks rule in general: re-thresholded to a matched benign false-positive rate the same 9 guards do leave sole blockers on the harmful side — 8 harmful prompts at the 3% target against 0 at the defaults (Table 1, walked in Appendix G) — so every degenerate interval and every ceiling in this section carries "at the vendor defaults" with it.

Blocker multiplicity. Table 21 gives, for each of the nine prompt groups across the four suites, the group size, the union block rate, the share of the blocked prompts carrying two or more blockers, and the mean of b, the number of the 9 guards blocking a prompt. Both quantities over blocked prompts are point estimates here. The intervals in the union column are a separate, simpler resampling

Table 17: The ablation table's blind spot, split by side and free of any credit denominator. Since $v = H + \lambda F$ is linear, $\phi_i(v) = \phi_i(H) + \lambda\phi_i(F)$; efficiency makes the harmful column sum to $H(N) - H(\emptyset) = 0.0000 - 1.0000 = -1.0000$, the whole harmful-side value of the stack. The leave-one-out column on that same side sums to 0.0000: every entry is exactly zero, because no harmful prompt has a sole blocker. Shapley therefore distributes all of a quantity of which ablation reports none, and unlike the shares of Table 20 this comparison does not depend on a normalisation — the ablation numerator is zero under every row of that table. Computed in exact rational arithmetic over all 512 coalitions; the harmful-side residual is zero, not rounded to it. Benign columns are shown at $\lambda = 1$ for completeness.

Guard	model	ϕ_i^H	LOO_i^H	ϕ_i^F	LOO_i^F
G	`granite-guardian-3.0-2b`	-0.1237	+0.0000	+0.2077	+0.1080
L	`Llama-Guard-3-8B`	-0.1068	+0.0000	+0.0045	+0.0000
S	`Llama-3.1-8B self-refusal`	-0.1186	+0.0000	+0.0133	+0.0000
M	`ShieldGemma-2B`	-0.0915	+0.0000	+0.1378	+0.0560
G8	`granite-guardian-3.0-8b`	-0.1233	+0.0000	+0.1239	+0.0280
Q	`Qwen3Guard-Gen-4B`	-0.1197	+0.0000	+0.0551	+0.0160
L4	`Llama-Guard-4-12B`	-0.0993	+0.0000	+0.0157	+0.0000
W	`WildGuard-7B`	-0.1158	+0.0000	+0.0036	+0.0000
L1	`Llama-Guard-3-1B`	-0.1015	+0.0000	+0.0143	+0.0000
sum		-1.0000	+0.0000	+0.5760	+0.2080

than the lattice's: a within-group item bootstrap at $B = 4000$, drawing prompts of that group with replacement and taking no account of the XSTest twin structure. It is not the cluster bootstrap of the bootstrap-design paragraph below, and for XSTest-safe — the one group here whose items are paired — it will read narrower than a clustered interval would. It is the same fact as Table 16 read per group, and inherits its operating point: the harmful groups place no mass at all at $b \leq 1$ in any suite, which in the table is the $b \geq 2$ column standing at its ceiling on all four of them while the union column stands at its own: every harmful prompt is blocked, and none of them by a lone guard. No benign group reaches that ceiling, and the main-suite benign group is furthest from it: of the XSTest-safe prompts the stack blocks at all, only 63.9 [55.9, 71.7]% are blocked by two or more guards, so a substantial minority of benign blocks are single-guard blocks and are therefore visible to ablation. Ablation sees the cost side and not the benefit side, and this table is the direct measurement of why. Mean b is the statistic that keeps discriminating once the guard set is large enough to saturate the share, but the saturated share is the load-bearing quantity, not a nuisance: it is what makes the harm column of the ablation table identically zero. Table 17 prices what that costs the instrument: the harmful halves of the exact Shapley values sum to the stack's whole harmful-side value while every harmful leave-one-out entry is zero, a comparison independent of how Table 20 normalises the shares, because its ablation numerator is identically zero.

Responsibility on each side of the loss. The multiplicity table says the two sides differ; the Chockler–Halpern degree of responsibility (Chockler & Halpern, 2004) says by how much. A guard blocking a prompt alongside k others takes $1/(1 + k)$ for it — k being the smallest number of other blockers that must be switched off before this one becomes pivotal — and Table 18 normalises those sums to a share within each side of the main suite. It is not an independent instrument, and we state the identity rather than leave it to be found: under any-one-blocks composition $1/(1 + k)$ is exactly the chance that a guard comes first among that prompt's blockers in a uniformly random order, so each column *is* the normalised Shapley value of that side's game — computed both ways they agree exactly, not approximately. It earns its place by making the asymmetry legible, not by corroborating it. On the harmful side the shares run from 0.0915 to 0.1237 against the 0.1111 a perfectly symmetric stack would give, a spread of 1.35×. On the benign side they run from 0.0063 to 0.3607, a spread of 57.7×, with granite-guardian-3.0-2b carrying the largest share and WildGuard-7B almost none. Harmful-side responsibility here is near-uniform and redundant; cost is produced by identifiable guards. That is the two-sentence version of this paper, and it is measured on the same decision matrices as everything else. The construction has a known limit worth stating: responsibility is defined per prompt from the blocker count alone, so it is symmetric by construction on any prompt every guard blocks, and the near-uniformity of the harmful column is partly that construction meeting

Table 18: Chockler–Halpern degree of responsibility (Chockler & Halpern, 2004), computed on the main suite and normalised to a share within each side. A guard blocking a prompt alongside k others takes $1/(1+k)$ for that prompt — k is the number of other blockers that must be switched off before it becomes pivotal — and the shares are those sums over the harmful and the benign pool separately. The two sides are the paper in one table. On the harmful side responsibility is near-uniform, every guard within a factor of 1.35× of every other and all 9 close to the 0.1111 a perfectly symmetric stack would give: harmful-side responsibility is near-uniform under this construction and no guard stands out as its source. On the benign side it is concentrated, a spread of 57.7× between granite-guardian-3.0-2b and WildGuard-7B: cost is produced by identifiable guards. The harmful column here reads a near-symmetry. These shares are not an independent instrument: on an OR game the $1/(1+k)$ rule is exactly the probability that a guard comes first among a prompt's blockers in a uniformly random order, so each column equals the normalised Shapley value of that side's game. We report it because it makes the asymmetry legible, not as corroboration.

Guard	model	harmful side	benign side
G	`granite-guardian-3.0-2b`	0.1237	0.3607
L	`Llama-Guard-3-8B`	0.1068	0.0079
S	`Llama-3.1-8B self-refusal`	0.1186	0.0232
M	`ShieldGemma-2B`	0.0915	0.2393
G8	`granite-guardian-3.0-8b`	0.1233	0.2151
Q	`Qwen3Guard-Gen-4B`	0.1197	0.0956
L4	`Llama-Guard-4-12B`	0.0993	0.0272
W	`WildGuard-7B`	0.1158	0.0063
L1	`Llama-Guard-3-1B`	0.1015	0.0249
uniform baseline	$1/\|N\|$	0.1111	0.1111
spread (max/min)		1.35×	57.7×

a saturated pool. That is the point rather than an artefact — a saturated pool is what a redundant stack produces — but it means the harmful column should not be read as evidence that the 9 guards are individually interchangeable on harmful traffic. Their single-guard pass rates are not (Table 3); it is the *union* that leaves nothing to divide.

Leave-one-in: the singleton sweep, at the ablation table's own budget. §5.2 adds the half of Shapley's sandwich the ablation habit omits, and Table 19 is the measurement behind it. The leave-one-in marginal is $v(\{i\}) - v(\emptyset)$, one execution per guard plus the empty coalition, 10 in total — the same budget as the ablation table and, on the guard stack, already paid: these are the singleton rows of Table 4 differenced against its empty row, which is to say the numbers a team produces when it benchmarks candidate guards before composing them. Against the exact Shapley ordering the leave-one-in ranking scores Spearman 0.9333, differing by the adjacent transpositions L1/Q, L4/Q, M/G.

Two notes on that comparison. First, the ablation column's own rank correlation with ϕ is 0.9129, *lower* than leave-one-in's, so the comparison runs our way, which is a reason to lean on it less rather than more. The rank statistic is the wrong summary of what leave-one-in buys: relative to the exact Shapley ordering, leave-one-out agrees on every guard pair it separates, and averaging the ranks of the 5 it ties at +0.000 prices that tie as if it were an ordering. What it buys is resolution inside that tie: {L, S, L4, W, L1} are separated by their singleton marginals and are not separable by any ablation at any budget, since the quantity being estimated there is identically zero. It is the measurement that distinguishes a duplicated guard from a null player, which §5.3 shows the cheap budgets otherwise cannot. Second, leave-one-in is biased, and in the opposite direction: it prices each guard in a stack of one, so a guard duplicating another is overstated exactly as badly as ablation understates its credit. Neither end of the sandwich is the credit; read against each other, the gap between them is the recapture signature, and that is the use we claim for it.

The credit share under six reporting conventions. Table 20 answers the question "what share of the credit do the guards ablation cannot see hold?" under three normalisations and three attribution or player-set conventions, because the answer moves by more than an order of magnitude across them and printing one alone invites the reader to take a convention for a measurement. The first three rows

Table 19: The other half of Shapley's sandwich, at the same price. $v(\{i\}) - v(\emptyset)$ is the leave-one-*in* marginal — the singleton block of Table 4, costing the same 10 executions as the ablation table and already run by anyone who benchmarked their guards before assembling them. Ranks are ascending, so rank 1 is the guard the column calls most helpful, and tied values share a rank. Against the exact Shapley ordering the leave-one-in ranking scores Spearman 0.9333, the adjacent transpositions being L1/Q, L4/Q, M/G. The ablation column scores 0.9129, *lower* than leave-one-in's, so the comparison runs our way, which is a reason to lean on it less rather than more. Relative to the exact Shapley ordering, leave-one-out agrees on every guard pair it separates. What it cannot do is order {L, S, L4, W, L1}, which it sends to a 5-way tie at +0.000 — a rank correlation on averaged ranks hides that failure rather than measuring it, and no budget removes it, since the quantity is identically zero. Leave-one-in separates them, being evaluated where redundancy is nil; it is biased the opposite way, overstating a duplicated guard as badly as ablation understates it, and prices each guard in a stack of one.

Guard	leave-one-in $v(\{i\}) - v(\emptyset)$	rank	Shapley ϕ_i	rank	leave-one-out LOO_i	rank
G	-0.6021	8	+0.0840	9	+0.1080	9
L	-0.8843	3	-0.1022	3	+0.0000	3
S	-0.9069	2	-0.1052	2	+0.0000	3
M	-0.5038	9	+0.0464	8	+0.0560	8
G8	-0.6881	7	+0.0006	7	+0.0280	7
Q	-0.8425	4	-0.0646	6	+0.0160	6
L4	-0.7916	6	-0.0836	5	+0.0000	3
W	-0.9490	1	-0.1122	1	+0.0000	3
L1	-0.8112	5	-0.0871	4	+0.0000	3
Spearman against the ϕ ordering	0.9333				0.9129	

hold the game and the zeroed set fixed and vary only the denominator. The efficiency row divides by $|v(N) - v(\emptyset)|$, which is what $\sum_i \phi_i$ equals to machine precision on this lattice, and gives 115.7%, above 100% because the mixed signs cancel in that denominator; dividing by $\sum_i |\phi_i|$ gives 71.5% and by the helpful mass alone 88.4%. Shapley is our primary attribution functional because it is efficient, but no denominator is canonical for a share of mixed-sign credit, so none of these ratios is primary: the denominator-free side decomposition of Table 17 is the primary comparison, and these rows are sensitivity displays. The last three rows change the attribution functional or the player set rather than the denominator: Banzhaf (Banzhaf, 1965; Dubey & Shapley, 1979) reweights the coalitions and gives 4.4%; the Owen value, which averages only over the orderings that keep each vendor's guards together — the provenance partition, 5 guard-model vendor blocs plus singleton S, fixed before any attribution was computed — gives 63.9% under the same $\sum_i |\phi_i|$ convention and 106.5% under the efficiency denominator, being efficient on this lattice to machine precision (residual $< 10^{-12}$); and holding S fixed — the only counterfactual a deployment can build — gives 3.6% over a zeroed set that is itself different, {L, L4, W, L1}. The spread across the six is far wider than the cluster-bootstrap interval on any one of them (71.5% [65.0, 76.5] on the second row), so the credit mass is not a stable quantity and no conclusion in this paper rests on it. Owen sharpens that, and against us rather than for us: it is the one row here that *reorders* the guards with every sign kept. In full, Owen ranks them W -0.167, S -0.156, Q -0.114, L -0.056, L1 -0.039, L4 -0.035, M +0.017, G8 +0.021, G +0.104, against Shapley's W, S, L, L1, L4, Q, G8, M, G: 6 of 9 change position, W takes the largest Owen credit of any player, and no guard changes sign (none). So the per-guard credit *ranking* is not stable across solution concepts either, and no claim in this paper rests on one. Because the full-player decision game is fixed, changing Shapley to Banzhaf or Owen changes neither the zeroed set nor the loss minimiser; those rows test only the allocation and ranking of coalition-averaged credit. The relation between minimiser and zeroed set is tested only where the game itself changes, in the matched-calibration and Qwen-convention controls (§5.1, Table 28, Appendix F).

Bootstrap design, in enough detail to reimplement. Uncertainty is a nonparametric cluster bootstrap over prompts, $B = 6000$ resamples, seed 20260904, percentile intervals at the 2.5th and 97.5th percentiles (Cameron et al., 2008). Each resample draws 513 clusters with replacement from the 513 main-suite clusters, rebuilds the prompt set from the drawn clusters, recomputes the full 512-coalition lattice on that prompt set, and recomputes ϕ, LOO, the loss gap and the zeroed

Table 20: How much credit the guards ablation cannot see receive under six defensible normalisation, attribution and player-set conventions. The first three rows are the same 9-player game and the same zeroed set {L, S, L4, W, L1}, divided three ways, and none is primary: mixed-sign values leave no canonical denominator for a share of credit. The *efficiency* row divides by $\sum_i \phi_i = v(N) - v(\emptyset)$, the smallest denominator once the signs cancel, so its ratio exceeds 100%; the denominator-free comparison is Table 17. The last three rows change the attribution rule or the player set rather than the denominator — Banzhaf reweights the coalitions, Owen restricts the permutations to those consistent with the provenance partition the guards actually come from (5 guard-model vendor blocs plus singleton S), and holding S fixed removes the 256 of 512 coalitions a team could never build — and the S-fixed row zeroes a different set, which is why its numerator is over {L, L4, W, L1}. Owen is the one row here that reorders the guards on the same 9-player game with every sign kept (§4.3), and it is efficient on this lattice to machine precision (residual $< 10^{-12}$), so 106.5% is its efficiency-normalised share against the 63.9% printed here. The spread across these six is an order of magnitude wider than the cluster-bootstrap interval on any one of them (71.5% [65.0, 76.5] on row two). The credit mass is therefore not a stable quantity and no conclusion in this paper rests on it; what every row on the full player set agrees on is which guards leave-one-out cannot see, since it does not depend on the weighting.

normalisation	zeroed set	numerator	denominator	ratio
efficiency, $\lvert v(N) - v(\emptyset)\rvert$	{L, S, L4, W, L1}	0.4904	0.4240	115.7%
$\sum_i \lvert\phi_i\rvert$ (the abstract's convention)	{L, S, L4, W, L1}	0.4904	0.6860	71.5%
helpful mass only, $\sum_{\phi_i<0} \lvert\phi_i\rvert$	{L, S, L4, W, L1}	0.4904	0.5550	88.4%
Banzhaf value, $\sum_i \lvert\phi_i^B\rvert$	{L, S, L4, W, L1}	0.0194	0.4415	4.4%
Owen value on the vendor partition, $\sum_i \lvert\phi_i^O\rvert$	{L, S, L4, W, L1}	0.4515	0.7069	63.9%
S held fixed, $\sum_i \lvert\phi_i\rvert$ on the 8 removable guards	{L, L4, W, L1}	0.0176	0.4883	3.6%

mass from it. For the intervals attached to named point-estimate objects the object is fixed before resampling: the loss-gap interval re-evaluates the point-estimate minimiser {W} on every draw and the zeroed-mass interval re-evaluates the point-estimate zeroed set {L, S, L4, W, L1}, so each interval says how the named subset or set varies under prompt resampling (re-selecting the set on each draw would put a different set's share into the interval). Separately, the argmin is recomputed inside every replicate, which is what the optimum-stability figure (81.3% at $\lambda = 1$) reports; selection optimism is handled by the split-half experiment of Table 31. That figure is a resampling statistic and not a held-out one, and the distinction is worth being explicit about: it says how often the argmin survives redrawing the clusters, not how a subset selected on one set of clusters scores on clusters it was not selected on. The second is the winner's curse, and nothing in this paragraph should be read as claiming the in-sample loss gap is free of it. It is not: the gap's minimising subset is selected on the same clusters the ratio is computed from, and that selection is priced separately — argmin chosen on half the clusters and its gap evaluated on the other half over 2000 splits gives 10.41 [5.18, 17.45] against 11.30 in sample, so the curse costs 7.9% of the gap (Table 31, §4.2).

Clusters are built as follows. StrongREJECT items are singleton clusters: each item is its own cluster. XSTest is not, because its safe and unsafe halves are constructed as contrastive twins — a safe prompt and an unsafe prompt sharing surface form — and resampling one without the other would break the contrast the suite exists to create and would understate the correlation between the two sides. XSTest's 18 prompt types collapse into 8 families by stripping the safe/unsafe marker from the type name, so that a safe type and the unsafe type built against it land in the same family. A family is not itself a cluster. A cluster is keyed by the pair (family, ordinal position of the item within its type), so one cluster is one contrastive twin group: a safe item and the unsafe item written against it at the same position within their respective types, or, in the two families that are triples rather than pairs — the discrimination family and the privacy family — the three items at that position. Six families therefore contribute one cluster per twin pair and two contribute one per triple, and each cluster's two or three items always resample together. Adding the 313 StrongREJECT singletons gives 513 clusters over the 763 main-suite prompts, which is the number the resampler draws with replacement. Clustering at the family level instead would give 8 XSTest clusters and does not reproduce 513. On JBB the cluster is the JBB-Behaviors `Index` field, which pairs each harmful behaviour with the benign behaviour matched to it by category, so each matched pair resamples as a unit; this is required for the paired multiplicity difference 4.27 [3.80, 4.75] to have a valid interval. Guards are never resampled: the 9 models are the population, not a sample from one, and the intervals we report are

Table 21: Blocker multiplicity and union block rate by prompt group, all four suites. b is the number of the 9 guards blocking a prompt. *Union blocked* is over the whole group; the last two columns are over the prompts the union actually blocks, since b is undefined where nothing blocked. The share at $b \geq 2$ is at its ceiling on every harmful group and on no benign group: at these guards' shipped operating points, removing any single guard leaves every harmful decision of the stack unchanged, while single-guard blocks survive on the benign side and stay visible to ablation. This is the per-group form of Table 16, where the same fact is stated per lattice as $H(N) = 0.0000$ with 0 sole-blocked harmful prompts and a minimum of 3, 5 and 2 blockers on the main, JailbreakBench and OR-Bench pools. The saturated share is the load-bearing quantity — it is what makes the harm half of every leave-one-out difference identically zero — while mean b is the statistic that keeps discriminating once a large guard set has pinned the share at its ceiling.

Suite	Group	n	union blocked (of group)	share $b \geq 2$ (of blocked)	mean b
harmful					
main	StrongREJECT	313	100.0% [100.0, 100.0]	100.0%	8.56
main	XSTest-unsafe	200	100.0% [100.0, 100.0]	100.0%	7.97
JBB	harmful	100	100.0% [100.0, 100.0]	100.0%	8.25
OR-Bench	toxic control	655	100.0% [100.0, 100.0]	100.0%	7.69
benign					
OR-Bench	hard-1k	1319	98.6% [98.0, 99.2]	97.3%	4.72
PHTest	controversial	500	96.4% [94.6, 98.0]	93.6%	4.42
JBB	benign	100	94.0% [89.0, 98.0]	79.8%	3.98
PHTest	harmless	500	79.2% [75.6, 82.8]	71.5%	2.79
main	XSTest-safe	250	57.6% [51.6, 63.6]	63.9%	2.34

over prompts only. Intervals therefore do not cover variation from re-drawing which guards a team happens to have deployed, and we make no claim about that distribution.

Source-stratified resampling. The cluster bootstrap above draws 513 clusters from one pool, so each replicate's mix of StrongREJECT and XSTest varies. Resampling the 313 StrongREJECT singletons and the 200 XSTest twin clusters as separate strata, so that every replicate keeps the suite's composition, changes no sign, minimiser or classification the paper reads, every per-guard interval endpoint moving by at most 0.0028: the full-over-{W} gap interval is [7.83, 18.34]$\times$ against the pooled 11.30 [7.84, 18.51]$\times$, and the zeroed set's share of $\sum_i |\phi_i|$ is [65.0, 76.4]% against 71.5% [65.0, 76.5] (`cluster_bootstrap_v1.py --stratify-source`).

C VALIDATING THE SELF-REFUSAL DETECTOR

Why this component needs its own validation. 8 of the 9 players are released models whose decisions are read from their own documented output convention. The last, S, is the generator's own refusal behaviour, and it is detected by a keyword matcher over the completion text: a fixed list of refusal substrings, and a prompt counts as blocked by S when one of them appears. S is one of the guards ablation sends to exactly zero, {L, S, L4, W, L1}, which carry 71.5% [65.0, 76.5] of $\sum_i |\phi_i|$, and it is the one player a deployment cannot switch off — which is why the S-fixed arm of §4.3 exists. A matcher that invented refusals would therefore corrupt the credit claim and the only counterfactual a team can actually build at once. (It is not in the unrestricted minimiser at this roster size.) We therefore validated it across three adjudication rounds totalling 470 items. The round sizes and per-stratum draws quoted below are design constants of the sampling protocol rather than results, but they are emitted as macros from the adjudication records like everything else, so that a re-drawn sample cannot leave the prose describing the old one.

Population and strata. The validated population is the main suite, 763 prompts in three groups, crossed with the matcher's own binary decision to give six strata (Table 22). Stratum membership is observed without error, since it is the matcher's deterministic output on stored text, which is what licenses the design below.

Table 22: Strata for detector validation, and round-one inclusion probabilities. The two harmful *passing* strata and the small benign blocking stratum were enumerated whole; the three large strata were subsampled. An error in a blocking stratum is a false positive (the matcher invents a refusal); an error in a passing stratum is a false negative.

Group	matcher	size	round-one n	inclusion prob.
StrongREJECT	blocks	296	33	0.111
XSTest-unsafe	blocks	181	33	0.182
XSTest-safe	blocks	15	15	1 (enumerated)
StrongREJECT	passes	17	17	1 (enumerated)
XSTest-unsafe	passes	19	19	1 (enumerated)
XSTest-safe	passes	235	33	0.140

Why inverse-probability weighting is required. The design deliberately oversamples the strata where an error would matter most and undersamples the large, easy ones, so inclusion probabilities differ by close to an order of magnitude across strata. A precision or recall computed by counting on the sample is therefore not an estimate of the population quantity: it is an estimate of a quantity under a distribution the design invented. Each adjudicated item is reweighted by the inverse of its inclusion probability (Horvitz & Thompson, 1952), and the reweighted confusion counts are the ones that describe the 763-prompt population: the Horvitz–Thompson weights estimate the four confusion-matrix totals, and precision and recall are ratios of those weighted totals. The difference is not cosmetic. On the round-one sample the raw recall of the matcher is 0.810, because the enumerated false-negative strata are, by design, a much larger share of the sample than of the population; reweighted over the round-one design, recall is 0.963 and precision is 1.000. Quoting the raw number would understate the matcher by a wide margin, and quoting it as if it described the population would be an error of design, not of arithmetic.

What the three rounds found. Round one adjudicated 150 stratified items under three differently-framed rubrics. It found 0 false positives and exactly 19 false negatives. The false-negative count is exact rather than estimated because both strata that can contain one — harmful prompts the matcher passed — were enumerated whole. Folding the corrections into the lattice left leave-one-out at exactly zero for both L and S and *widened* the λ window over which {W} is optimal. Round two adjudicated 200 further blocked items, disjoint from round one, drawn equally from the two large harmful blocking strata, again under three framings, and again found no false positive; pooled over both rounds the matcher has 0 false positives in 281 adjudicated blocked items. After round one the 95% Wilson upper bound on the *per-stratum* false-positive rate was 10.4% (Wilson, 1927); pooled over the 281 adjudicated blocked items of both rounds it is 1.35%, and the finite-population bounds for the two large blocking strata are 1.35% and 1.10% at sizes 296 and 181.

Round two also produced the result that decided where round three had to go. A pessimistic stress column that flips only the *harm*-side unadjudicated tails reproduces the corrected lattice exactly: harm-side label pessimism is absorbed by guard redundancy, for the same reason every harm-side leave-one-out difference is zero. Every remaining tooth in the pessimistic collapse came from one untested stratum, the benign prompts the matcher passes, where the bound was still loose enough to fabricate a large number of imaginary false refusals. Round three therefore adjudicated 120 items from that stratum alone, again disjoint from both earlier rounds and again under three framings, one of them written specifically to hunt refusals carrying no refusal keyword. It found no missed over-refusals. 153 of 235 items in that stratum have now been read, with 82 unexamined; the pooled 95% Wilson upper bound on the miss rate is 2.45% and the finite-population bound is 0.85%, rising to 4.64% under the harshest reading, in which both items ever marked `unclear` are counted as misses. Round three added no missed over-refusal, which is the error it was designed to hunt. It did produce the single false positive in the corrected column: the one item all three framings marked `unclear`, described below, which the matcher blocks under the current decision records and which all three framings read as compliance. That item is the difference between the 19 harm-side corrections of round one and the 20 the lattice applies. What round three changed most is the pessimistic column, whose benign tail now runs on a measured bound instead of an assumed one. Under that column S's leave-one-out is no longer exactly zero (+0.004 at $\lambda = 1$) and the {W} optimality window retains

83.0% of its corrected span, or 70.8% under the harshest reading. L's leave-one-out is exactly zero in every column. We report this rather than the corrected column alone: the residue is small enough that no conclusion in the paper turns on which column is quoted, and it is not zero.

The same stress on the 9-player lattice. The columns above run on the three-player validation lattice. The headline zeroed set is read off the 9-player game and S is a member of it, so we re-enumerate all 512 coalitions with S's column replaced by each stress column and every other decision held at the stored record (`detector_stress9_v1.py`). The primary pessimistic column changes 8 S decisions, the harsh unclear-as-miss column 10. Under both, every harmful-side full-stack leave-one-out entry stays exactly zero and no harmful prompt acquires a sole blocker. The primary column leaves the total-loss zeroed set at {L, S, L4, W, L1}, carrying 71.7% of $\sum_i |\phi_i|$; the harsh column moves S's row to +0.004, leaving {L, L4, W, L1} at 57.3% with the full-stack loss at 0.580. The λ=1 minimiser is {W} under the primary column and {W} under the harsh one, at 11.30× and 11.38× against the corrected 11.30×. Detector uncertainty moves the secondary total-loss zero set and the credit-mass illustration; it does not touch the harmful-side support collapse.

The labels are LLM-adjudicated, and the agreement statistics are uninformative. No human adjudicated any item. Every label in all three rounds is another model's judgement, taken under three differently-framed rubrics. That is the single largest caveat on the S player and it applies to all three rounds. It also has a specific consequence for how the rounds may be reported. In round one the three framings genuinely disagreed on a handful of items — three-way agreement 96.0%, pairwise Cohen's κ as low as 0.908, one pair at 1.000 (Cohen, 1960) — and that round's agreement statistics carry some information. In round two the three framings returned a literally constant label vector, and in round three they returned vectors differing on 1 item out of 120 (119 `comply` and 1 `unclear`). On a near-constant vector κ is degenerate: its expected-agreement term collapses and the statistic stops measuring anything about independence. We therefore do not quote later-round agreement as evidence that the framings read independently. The only evidence of independent reading in those rounds is that the adjudicators' free-text notes differ byte-for-byte, which is weak evidence and we treat it as such. The one `unclear` item in round three was read directly and is a knowledge hedge about a film plot with no safety register in it; the one in round one is a clarifying question. Neither is a missed refusal.

Every figure in this section is derived. The detector figures quoted above are recomputed from the sample files and the three round outputs by the detector derivation script in the release, so a re-drawn sample cannot leave the prose describing the old one. The corrected lattice applies all 20 adjudicated corrections — the 19 false negatives recovered on harmful prompts and the one false positive on a benign one — and every lattice quantity in this paper is computed after all 20.

What remains unvalidated. Four limits, none of which the bounds above remove. Every label is a model's judgement, so all of the above bounds the matcher against an adjudicator, not against the truth. 82 items in the benign passing stratum, and the unadjudicated tails of the two large blocking strata, were never read; the bounds are bounds, not proofs of zero. The adjudicators read each response truncated to 300 characters, and a response that runs to the full 300 characters may have been cut mid-turn: 112 of the 120 round-three responses and 25 of the 200 round-two responses reach that width, so in those a post-cap pivot from compliance to refusal would be unobservable — though the matcher scored the same truncated text, so the measurement and the artefact agree. And all of it is conditioned on one generator's refusal style and on two prompt distributions; none of it transfers to another model or another benign set. The correction the validation produced strengthens the claims it touches rather than weakening any of them, which is the direction that should make a reader more rather than less suspicious, and is why we give the design and the unexamined remainder here in full.

D OPERATING-POINT SWEEP

Every guard decision in the paper is taken at its vendor's documented default operating point, and the obvious objection is that the inversion is an artefact of those defaults. Because we store the decision-token log-probabilities behind every decision (Appendix F), the objection is answerable offline: a coalition table at a different operating point is a re-read of stored scores, not a re-run of 9 models.

Table 23: Operating-point sweep over the full 9-guard lattice: at each threshold all 512 coalitions are re-derived from stored decision-token scores, so no model is re-run, and λ is held at 1. The minimiser is {W} at three of the 5 thresholds, with two exceptions ({S} at 0.1, {L,W} at 0.9); it is drawn entirely from that threshold's own zeroed set at four, the exception being 0.3, where the minimiser is not one of the guards that threshold zeroes. The full stack's gap over the minimiser runs from 3.39× to 12.09×. At the most permissive threshold the guards that strictly reverse sign, meaning $\phi_i < 0$ against a leave-one-out above 10^{-3}, are G, M, Q, L4, L1; the remaining guards L, S, G8, W do not reverse but stay at or within 2×10^{-4} of zero, S at +0.0002 against $\phi_S = -0.1113$. An exact zero and a sign disagreement are different failures and are counted separately here. The minimiser at the most permissive threshold, {L,W}, is a different object from {S,W}, the minimiser below the λ window.

thr.		G	L	S	M	G8	Q	L4	W	L1	full-stack	best	minimiser
0.1	ϕ_i	+0.291	-0.100	-0.102	+0.044	+0.086	-0.041	-0.086	-0.095	-0.077	0.920	0.093	{S}
	LOO_i	+0.1640	+0.0000	+0.0000	+0.0120	+0.0040	+0.0040	+0.0000	+0.0000	+0.0000			
0.3	ϕ_i	+0.213	-0.101	-0.104	+0.031	+0.048	-0.051	-0.085	-0.104	-0.083	0.764	0.063	{W}
	LOO_i	+0.1680	+0.0000	+0.0000	+0.0120	+0.0320	+0.0200	+0.0000	+0.0040	+0.0000			
0.5	ϕ_i	+0.084	-0.102	-0.105	+0.046	+0.001	-0.065	-0.084	-0.112	-0.087	0.576	0.051	{W}
	LOO_i	+0.1080	+0.0000	+0.0000	+0.0560	+0.0280	+0.0160	+0.0000	+0.0000	+0.0000			
0.7	ϕ_i	-0.010	-0.102	-0.107	+0.021	-0.044	-0.073	-0.085	-0.114	-0.086	0.400	0.053	{W}
	LOO_i	+0.0480	+0.0000	+0.0000	+0.0640	+0.0240	+0.0200	+0.0000	+0.0000	+0.0040			
0.9	ϕ_i	-0.061	-0.104	-0.111	-0.009	-0.100	-0.099	-0.081	-0.115	-0.080	0.240	0.071	{L,W}
	LOO_i	+0.0240	+0.0000	+0.0002	+0.0501	+0.0000	+0.0120	+0.0080	+0.0000	+0.0080			

The sweep is over the full 9-guard lattice: at each of five thresholds on p_{unsafe}, from 0.1 to 0.9, the stored decision-token scores are re-thresholded, the decision matrix is rebuilt, and all 512 coalitions are re-enumerated. Thresholding at 0.5 reproduces the vendor-default decisions exactly, which is both a row of the sweep and a check on the decision parser.

One player is held fixed and one moves too little to matter. S is a keyword matcher over free text and exposes no score to threshold, so its column carries its keyword decision unchanged in every row of the sweep; the 8 scored guards are the ones that move. Of those, L4 barely does: 92.5% of its rows have a p_{unsafe} within 10^{-3} of 0 or 1, against 63.0% for L and less for the rest, so every interior threshold reproduces almost exactly the same L4 decisions. L4 is in the sweep, but the sweep is close to uninformative about *its* operating point, and we read nothing from its rows about how L4 would behave at a genuinely different one.

Table 23 gives, at each threshold and at $\lambda = 1$, both halves of the comparison: a ϕ_i row and a LOO_i row across all 9 guards, followed by the loss of the full stack, the loss of the best coalition, and which coalition that is. Every per-guard value quoted in the rest of this section is a cell of that table. Leave-one-out is printed there to four decimal places rather than three because at the most permissive threshold S sits below what a third place can show, and the distinction between an exact zero and a small nonzero score is one this section turns on. What the table does not carry is the 512-coalition lattice behind each row; those are in the released sweep lattices (Appendix F).

The inversion survives every threshold tested, though the minimiser does not stay put: it is {W} at three of the five thresholds, the exceptions being {S} at 0.1, {L,W} at 0.9. It is drawn entirely from that threshold's own leave-one-out-zeroed set at four of the five; the exception is 0.3, where the minimiser is not among the guards that threshold zeroes. The loss gap over the full stack runs from 3.39× to 12.09× throughout. The optimum at the most permissive threshold, {L,W}, is a different object from {S,W}, the optimum below the λ window of §4.2: here the guards' operating point moves and λ is held at 1, not the reverse.

The set of guards ablation zeroes is exactly {L, S, L4, W, L1}, unchanged from the vendor default, at 2 of the 5 thresholds (0.1, 0.5); at the others it changes: at 0.3 and 0.7 it loses one member, and at the most permissive threshold it loses S, L4 and L1 and *gains* G8, whose leave-one-out is exactly zero only there.

A common threshold is not, however, the right control for the objection it is answering. Moving 8 differently-calibrated models to the same cut on p_{unsafe} preserves the vendor calibration mismatch rather than removing it; the control that removes it matches each guard to a common benign false-positive rate instead, and it is in Appendix G, where it shrinks the effect this section reports as robust.

At the most permissive threshold every one of the 9 guards carries a negative Shapley value — on the lattice all 9 reduce loss — and not one carries a negative leave-one-out score. Two different failures are stacked there, and they have to be reported apart. The first is the loss of resolution: {L, S, G8, W} still sit at or within a rounding step of zero under ablation, L at +0.000 against a Shapley value of -0.104, G8 at +0.000 against -0.100, and S not exactly at zero but at +0.0002 against -0.1113, which is a rounding step and not a correction. The second is sign disagreement, and it appears in the other 5 players, {G, M, Q, L4, L1}: +0.024 against -0.061 for G, +0.050 against -0.009 for M, +0.012 against -0.099 for Q, +0.008 against -0.081 for L4, and +0.008 against -0.080 for L1, so ablation nominates as most disposable 5 guards the lattice scores as helping.

That count is a *raw-sign* one, and we label it as such rather than let it pass for the stricter thing. Under the single rule the strict set is {Q} and {G8} is unsupported rather than contradicted, and G — the largest ablation score, so the first guard any pruning pass removes — keeps a Shapley value 0.78 of its own row. §4.1 applies a single rule — a strict disagreement requires both the ϕ and the leave-one-out interval to exclude zero — and that rule needs intervals, which the cluster bootstrap supplies only for the full-stack lattice at the vendor-default operating point. It is not available for the re-thresholded lattices here, or for the drop-one lattices of Appendix G, and we did not resample either. So every disagreement count outside Table 14, in this section and in Tables 26, 27 and 32, counts $\phi_i < 0$ against a leave-one-out above 10^{-3} and nothing more. Where the intervals do exist, the single rule gives 1 strict disagreement, {Q}, against the raw count of 1, with {G8} unsupported rather than contradicted; the raw counts elsewhere should be read as upper bounds on what the interval rule would return.

It is worth being exact about which guards those are, because it is not the same set as before. The guards ablation zeroes at the vendor-default operating point are {L, S, L4, W, L1}; of those, {L, S, W} do not reverse at the permissive threshold — they stay at zero, where ablation was already silent about them — and 2 cross, {L4, L1}. The other 3 reversing guards, {G, M, Q}, are ones ablation already scored as positive at the default operating point — as actively harmful — and it scores them that way again here while the lattice scores all 3 as helping. The permissive threshold therefore does not turn the exact zeros into signs opposite to ϕ; it leaves the zeros where they were and adds a second, worse failure mode beside them. We repeat the caveat from §6 here because this is where the reversals live: recapture drives leave-one-out toward zero and cannot by itself carry it across zero. The crossing requires per-query weights that are not sign-definite, which the mixed-sign $H + \lambda F$ supplies and a harm-side-only objective would not. Reversal is a property of this loss together with redundancy, not of redundancy alone.

E THE FIVE-GATE ABSTENTION CASCADE

The system. The second system is a multi-source question-answering cascade over Korean public-health data — a statistics database, a corpus of legal text, and a knowledge graph — with five abstention gates: insufficient evidence (E), privacy restriction (P), out-of-schema request (S), missing graph path (G), and low router confidence (R). Any gate that fires withholds the answer, so the composition rule is the same any-one-blocks rule as the guard stack, over components that share neither training corpus nor architecture and are mostly not learned at all. Full 2^5 lattices were enumerated on both sets. The value function is selective risk: the error rate among the questions the cascade does answer.

The held-out set. The development set is 60 hand-written questions, which were used for diagnosis, so measurements on it are not out of sample. The replication set is 150 questions built mechanically: question text is generated from templates over slots drawn from the stores, and the label — answerable, or which gate *should* fire — is decided by a store lookup rather than by judgement, so a year outside the statistics range, an article number absent from the legal corpus, or a missing graph edge determines the gold label by construction. The generator never imports the system under test, which is what keeps the labels from being the system's own opinion. Exact string overlap with the development questions is asserted to be empty at build time.

Results. Table 24 reports both sets side by side in two panels. The upper panel is selective prediction — risk, AUROC and AURC with abstention off and with the full cascade on — and the lower panel is the per-gate attribution on each lattice, every gate's ϕ_i and LOO_i, so the count of exactly-zero

Table 24: The five-gate QA abstention cascade, development set and mechanically-labelled held-out set. The held-out set shares no exact prompt string with the development set. The lower panel is the per-gate attribution on each lattice, so the count of exactly-zero leave-one-out scores in the last row is checkable from the table: the same three gates — P, S and G — have identically zero leave-one-out differences on both lattices, while none of their Shapley values is zero on either.

	development ($n = 60$)		held out ($n = 150$)	
selective prediction				
risk, abstention off		0.267		0.267
risk, abstention on		0.114		0.128
AUROC, abstention off		0.878		0.825
AUROC, abstention on		0.759		0.740
AURC, abstention off		0.114		0.137
AURC, abstention on		0.146		0.166
per-gate attribution				
gate	ϕ_i	LOO_i	ϕ_i	LOO_i
insufficient evidence (E)	-0.074	+0.014	-0.065	+0.018
privacy restriction (P)	-0.015	+0.000	-0.009	+0.000
out-of-schema (S)	-0.032	+0.000	-0.034	+0.000
missing graph path (G)	+0.004	+0.000	-0.021	+0.000
low router confidence (R)	-0.036	+0.012	-0.009	+0.006
gates with LOO exactly zero		3 of 5		3 of 5

leave-one-out scores in the final row can be checked against the rows above it instead of being taken on trust. What the table does not carry is the traffic behind those numbers: how many questions each gate abstains on in the full cascade, how many of those the remaining gates recapture, and by which gate. Those are in the released cascade lattice (Appendix F), and the facts we draw from them are stated in the text below.

The abstention machinery replicates in both directions it is supposed to move. Selective risk falls from 0.267 with all gates off to 0.128 with the full cascade. The ranking quality of the retained confidence signal falls with it, AUROC 0.825 to 0.740 and AURC 0.137 to 0.166, because abstaining truncates the execution that produces the signal (El-Yaniv & Wiener, 2010): a system that declines to run cannot also rank what it declined to do. The risk–coverage tradeoff itself is standard (El-Yaniv & Wiener, 2010); the truncation mechanism is our own reading of the traces, and we have established it on the development set rather than on the held-out set.

Half of the attribution pathology replicates here and half does not. What replicates is the invisibility of the gates to ablation: 3 of the five gates — P, S and G — have every one of their abstentions recaptured by a gate that remains, so with any one of them removed the set of questions answered is identical to the full cascade's and their leave-one-out scores are exactly zero. All three carry nonzero Shapley value on the held-out lattice, and the same three gates are the zeroed ones on the development lattice. On the held-out lattice every gate's Shapley value is negative — each reduces risk — while on neither lattice is any gate's leave-one-out score negative, so an ablation table of this cascade credits none of its five gates with reducing risk. Four of the five carry a negative Shapley value on the development lattice too; the exception is the missing-graph-path gate G, which is slightly positive there and negative on the held-out set, and which we therefore do not claim as a risk-reducing gate on either lattice alone. The efficiency residual is zero, as on the guard stack. Uncertainty on the development lattice is an item bootstrap, $B = 10000$; the held-out lattice is reported as point values without an interval, which is a limitation and we do not describe those point values as significant.

What does *not* replicate is the best-subset half. Under the cascade's own value function — selective risk on the questions it answers — the minimiser of the held-out 2^5 lattice is {S,G,R} at risk 0.110, and on the development lattice it is {S,R} at 0.095 (2 coalitions tie at the held-out minimum and 2 at the development one; we quote the smallest of each). Both minimisers contain R, the low-router-confidence gate, whose leave-one-out is not zero. So on this system the minimiser is *not* drawn entirely from the guards ablation zeroes, which is the claim the guard stack supports and this cascade does not. We state the narrow claim only: three of five gates score leave-one-out exactly zero while

carrying nonzero Shapley value, and the ablation table of this cascade credits none of its five gates with reducing risk.

One weaker comparison does hold and is worth having, because it is the practitioner's version of the question. The single-gate coalition {E} answers 98 of the 150 held-out questions at risk 0.122, against 78 at 0.128 for the full five-gate cascade. It answers more and errs no more often, and every error the full cascade makes is also made by {E}: the full cascade's error set is a subset of {E}'s. The four gates the cascade adds on top of E prevent 2 of {E}'s 12 errors, and pay for those two by declining 20 more questions — which is why its risk among answered questions is no lower. That says the full cascade is beaten by a much smaller subset; it does not say the beating subset is one ablation points at, and {E} is not — E's leave-one-out is the largest of the five.

F REPRODUCTION NOTES

Nothing here requires re-running a model. All 8 guard models are publicly released checkpoints, run at their vendors' documented default operating points under each vendor's own output convention and greedy decoding. We store one JSONL record per (prompt, guard) pair, containing the prompt id and group, the guard name and the exact model identifier, the operating point in force, the guard's raw output string, the parsed block decision, the log-probabilities of the two decision tokens, the resulting p_{unsafe}, the probability mass in the two decision poles, the position at which the decision token was found, and a flag recording whether it was found at all — so a decision that could not be parsed under the vendor's convention is visible in the record rather than silently coerced. One guard is three-tiered rather than two: Q emits Safe/Unsafe/Controversial, and we group Controversial on the block side, so its p_{unsafe} is $1 - P(\text{Safe})$ and its record keeps the per-label log-probability so the tier boundary can be moved offline. Two consequences we state rather than let a reader find. The scored decision and the greedily decoded label disagree on 2 of Q's 3937 rows, where the grouped mass sits just above the threshold while Safe wins the argmax; and on 59 rows two pole log-probabilities are bit-identical, because bf16 quantises them together, which greedy decoding breaks in vocab-id order and our conditional breaks in label order. What holds on all 3937 rows is that the decoded label *is* the argmax pole, and that is what `-selftest` asserts for this guard; the stricter agreement it once asserted universally is true only of the two-pole guards. The same quantisation puts 30 main-suite rows across 7 guards at p_{unsafe} of exactly 0.5, which the strict vendor threshold passes; 21 of them are benign, and resolving all of them to block would raise the full stack's benign block rate from 0.5760 to 0.6000 and leave $H(N)$ at zero. The self-refusal player has no decision token, so its score fields are null; its record carries the generator's own reply verbatim and the matcher's verdict on it, which is what the adjudication rounds of Appendix C re-read.

Checkpoints and revisions. The generator behind S is the `NousResearch/Meta-Llama-3.1-8B-Instruct` hub mirror; we did not verify it byte-for-byte against Meta's gated repository. The hub commit of every model and dataset the stored records came from is recorded in `revisions_v1.json`, the inference script loads each model at that commit and each dataset from that snapshot, refusing an unpinned repository unless told to, and the run environment's library versions are recorded beside them.

Moving that tier boundary, since we said it could be moved. Grouping Controversial with Unsafe is the vendor's own tier order, but it is a choice, and it is the one convention in this paper that a reader could reasonably set differently. Blocking on Unsafe alone passes 71 main-suite prompts Q previously blocked. The half of the paper that is about the harmful side does not move: $H(N)$ stays at 0.0000, no harmful prompt gains a sole blocker (0), and the harmful leave-one-out column is still identically zero. The half about subsets does move, and against us. Benign block rate falls from 0.5760 to 0.5600 and the loss gap from 11.30× to 10.98×, but the zeroed set turns over: Q enters it and W leaves, so the set is {L, S, Q, L4, L1} and the loss-minimising coalition {W} is no longer drawn from it. This is the second control to separate the two halves that way — the matched-FPR lattices of §5.1 were the first — and the two halves come apart differently. The harmful half survives this convention and not matched calibration, where 8 harmful prompts regain a sole blocker (Table 1); the best-subset half survives neither. Both are properties of the operating points and the decision conventions, not of any-one-blocks composition itself.

Every table in this paper is a function of those records. The coalition lattices are a Boolean OR over guard columns; the Shapley values are exact enumeration over the resulting value function; the operating-point sweep re-thresholds the stored p_{unsafe} before the OR. A reader who wants to check the arithmetic needs the stored decision matrices and none of the models. A reader who wants to check the decisions themselves needs the 9 checkpoints; we ship a small smoke subset for that purpose, so that a freshly fetched checkpoint can be compared against stored decisions on eight prompts before committing to a full re-run.

Artifact structure. The release is organised as: raw per-guard decision records, one file per guard per suite; the lattice enumerator, which reads those files and writes a coalition table; the exact Shapley and leave-one-out computation; the cluster bootstrap, with the cluster definitions of Appendix B and seed 20260904; the stratified sample builders and the adjudication records for the three detector rounds, including the inclusion probabilities of Table 22, which are needed to reproduce the reweighted rates and not merely to document them; the five-gate cascade's pre-computed lattice and selective-prediction artifacts; the script that derives every detector figure of Appendix C from the adjudication records; the control script that produces every table of Appendix G; and the table generator that emits every numbered table in this paper from those artifacts. All result numbers in the paper are generated into a single macro file and referenced by name in the source, so no number is typed by hand in more than one place.

Every suite we use is publicly available (Souly et al., 2024; Röttger et al., 2024; Chao et al., 2024; Cui et al., 2025; An et al., 2024), and the decision records carry the prompt text under each suite's licence, which the release lists — except for the 92 StrongREJECT prompts its authors collected from earlier datasets, which keep their original licences, several of them unstated. For those the records carry the prompt's SHA-256 instead, and a shipped script restores the text from the official StrongREJECT file. No number in this paper reads a prompt string, and the release reproduces every one without them. The Korean public-health cascade depends on data sources whose redistribution we cannot grant; for that system we release its pre-computed coalition lattice and the selective-prediction summaries behind the interval, from which every number in Appendix E is read and the Shapley and leave-one-out columns recomputed; not its per-question Appendix E but not to rebuild the retrieval stores.

G Additional robustness controls, including the ones we lose

This appendix reports the robustness controls, the ones that go against us first: three perturbations that each remove part of the headline, and three out-of-sample and convention controls — a held-out matched false-positive rate, a split-half check on the subset the loss gap is selected from, and the Owen value on the provenance partition (Table 31). Nothing here re-runs a model: each control is a re-read of the same stored decision matrices, so any of them can be recomputed in seconds and none of them depends on our being right about anything.

A cheap greedy procedure finds the minimiser, so enumeration is not required. The ablation table does not point *away* from the good subset. Leave-one-out elimination run to a fixed point — which is what a team does after reading an ablation table, rather than reading it once and stopping — reaches the exact global minimiser on all 3 suites at every λ in Table 25, in 46 coalition reads at the equal-weighting point on the main suite against 512 for the full lattice. What survives is narrower and still worth the paper: a single ablation table is uninformative about which guards produce the screening, and exact-zero rows are not evidence that those guards can be deleted together. Finding the good subset is cheap. Reading the ablation table once and deleting what it zeroes is what fails. Greedy *forward* selection from the empty coalition is not a substitute — it is cheaper still, 18 reads at the equal-weighting point, and it misses the minimiser on OR-Bench at the lowest λ in the grid — which is the asymmetry worth carrying: the procedure that starts from the stack you already have is the one that works.

Self-refusal cannot be removed from a deployment, so the headline must be reported with it held fixed. S is the generator's own refusal behaviour. No team can ship the coalition that excludes it, so the unrestricted game credits a counterfactual nobody can build, and the credit-mass headline taken alone overstates what the restriction leaves. Table 26 plays the game on the 8 removable guards

Table 25: Greedy subset search against the exact minimiser, on all 3 suites and at every λ the paper reports. *Backward* is leave-one-out elimination run to a fixed point — the procedure a team actually runs after reading an ablation table — and *forward* is greedy addition from the empty coalition. *exact* records whether the returned set is the lattice argmin. *reads* counts coalition evaluations, against 512 for the full lattice. Backward elimination is exact in every row; forward selection is not. This is the control that withdraws any claim that the good subset requires enumeration: it does not, and one more round of elimination past the ablation table finds it.

suite	λ	argmin	backward	exact	reads	forward	exact	reads
main	0.25	{S,W}	{S,W}	yes	45	{S,W}	yes	25
main	0.5	{S,W}	{S,W}	yes	45	{S,W}	yes	25
main	1.0	{W}	{W}	yes	46	{W}	yes	18
main	1.5	{W}	{W}	yes	46	{W}	yes	18
main	2.0	{W}	{W}	yes	46	{W}	yes	18
jbb	0.25	{L}	{L}	yes	46	{L}	yes	18
jbb	0.5	{L}	{L}	yes	46	{L}	yes	18
jbb	1.0	{L}	{L}	yes	46	{L}	yes	18
jbb	1.5	{L4}	{L4}	yes	46	{L4}	yes	18
jbb	2.0	{L4}	{L4}	yes	46	{L4}	yes	18
orbench	0.25	{L,S,L4}	{L,S,L4}	yes	43	{W}	no	18
orbench	0.5	{L,L4}	{L,L4}	yes	45	{L,L4}	yes	25
orbench	1.0	{L}	{L}	yes	46	{L}	yes	18
orbench	1.5	{L4}	{L4}	yes	46	{L4}	yes	18
orbench	2.0	{L4}	{L4}	yes	46	{L4}	yes	18

Table 26: The unrestricted lattice beside the only counterfactual a team can actually build. S is the generator's own refusal behaviour and cannot be removed from a deployment, so the right-hand block holds S in every coalition and plays the game on the 8 removable guards. Holding S fixed shrinks the zeroed set to {L, L4, W, L1} and its share of $\sum_i |\phi_i|$ to 3.6%, and leaves no guard disagreeing in sign at any λ. The full-stack loss is identical in the two blocks — fixing a player changes which coalitions are available, not the losses — so the gap differs exactly where the unrestricted minimiser EXCLUDES S, since the restriction must then add it back, at $\lambda \in \{1.0, 1.5, 2.0\}$, and agrees elsewhere. *dis.* is a *raw-sign* count — guards with $\phi_i < 0$ against a leave-one-out above 10^{-3} — because the cluster bootstrap exists only for the full-stack main-suite lattice and these rows have no intervals. On that lattice the single rule of §4.1, that a strict disagreement needs *both* intervals to exclude zero, gives 1 ({Q}) which at this roster size coincides with the raw count, leaving {G8} unsupported rather than contradicted. *share* is the $\sum_i |\phi_i|$ convention throughout; Table 20 gives that set under the other conventions, and a smaller one once S is held fixed.

	all removable					S held fixed				
λ	minimiser	gap	zeroed	share	dis.†	minimiser	gap	zeroed	share	dis.†
0.25	{S,W}	6.58	{L,S,L4,W,L1}	61.8	4	{W}	6.58	{L,L4,W,L1}	6.9	0
0.5	{S,W}	7.22	{L,S,L4,W,L1}	72.5	4	{W}	7.22	{L,L4,W,L1}	4.3	0
1.0	{W}	11.30	{L,S,L4,W,L1}	71.5	1	{W}	7.59	{L,L4,W,L1}	3.6	0
1.5	{W}	15.16	{L,S,L4,W,L1}	53.6	1	{W}	7.72	{L,L4,W,L1}	3.6	0
2.0	{W}	18.29	{L,S,L4,W,L1}	41.8	1	{W}	7.79	{L,L4,W,L1}	3.7	0

with S present in every coalition. The zeroed set falls to {L, L4, W, L1}, its share of $\sum_i |\phi_i|$ falls to 3.6% — the same convention as the unrestricted 71.5%, and the last row of Table 20, where it is the smallest of the six figures that sentence could carry — and the raw sign disagreements fall to 0. We report this column beside the unrestricted one throughout rather than choosing between them, because they answer different questions: the unrestricted column says how the coverage is produced, the fixed column says what a team can act on. What does not change under the restriction is the lattice itself — the minimiser is still inside the zeroed set at every λ, and the full stack still loses to it by $7.59\times$ at $\lambda = 1$.

No single guard carries the effect. The natural objection is that one over-eager guard makes the others redundant by construction, and that the whole result is an artefact of including it. The obvious suspect is G, which blocks more benign prompts than any other guard in every benign group (Table 13). Table 27 rebuilds the lattice with each guard removed in turn, 10 further complete lattices in all. Removing the suspect does not dissolve the effect: 64.3% of the credit still sits on guards ablation cannot see, at a gap of $9.18\times$ — every share in that table being the $\sum_i |\phi_i|$ convention of

Table 27: Robustness of the zeroed set to the guard population itself, at $\lambda = 1$. Each row rebuilds the lattice on a smaller guard set and recomputes everything. *drop G* removes the guard with the highest benign block rate, the obvious candidate for the objection that one over-eager guard makes the others redundant by construction. It does not carry the effect: removing it still leaves 64.3% of the credit on guards ablation cannot see. Across all 9 single drops that share never falls below 62.4% and the zeroed set never shrinks below 4 guards; only dropping M raises it. *one per vendor* keeps one guard model from each of the 5 vendors and the assistant's own refusal as a singleton. The worst share over all 9 single drops is 62.4% and the worst gap 7.05×, on different rows; no single guard carries the effect. *dis.* is a *raw-sign* count — guards with $\phi_i < 0$ against a leave-one-out above 10^{-3} — because the cluster bootstrap exists only for the full-stack main-suite lattice and these rows have no intervals. On that lattice the single rule of §4.1, that a strict disagreement needs *both* intervals to exclude zero, gives 1 ({Q}) which at this roster size coincides with the raw count, leaving {G8} unsupported rather than contradicted. *share* is the $\sum_i |\phi_i|$ convention throughout; Table 20 gives that set under the other conventions, and a smaller one once S is held fixed.

guard set	minimiser	all-guard	best	gap	zeroed	share	dis.†
all 9	{W}	0.576	0.051	11.30	{L, S, L4, W, L1}	71.5	1
drop G	{W}	0.468	0.051	9.18	{L, S, L4, W}	64.3	2
drop L	{W}	0.576	0.051	11.30	{S, L4, W, L1}	69.0	2
drop S	{W}	0.576	0.051	11.30	{L, L4, W, L1}	68.7	2
drop M	{W}	0.520	0.051	10.20	{L, S, L4, W, L1}	74.0	1
drop G8	{W}	0.548	0.051	10.75	{L, S, L4, W, L1}	69.0	1
drop Q	{W}	0.560	0.051	10.98	{L, S, L4, L1}	62.4	2
drop L4	{W}	0.576	0.051	11.30	{L, S, W, L1}	69.8	2
drop W	{L,S}	0.576	0.082	7.05	{L, S, L4, L1}	68.8	2
drop L1	{W}	0.576	0.051	11.30	{L, S, L4, W}	69.2	2
one per vendor	{W}	0.548	0.051	10.75	{L, S, W}	66.5	1

Table 20, computed on each row's own smaller game. Across all 9 single drops that share stays between 62.4% and 74.0%, the zeroed set never falls below 4 guards, and the gap never below 7.05×; reducing to one guard per bloc gives 66.5% at 10.75×.

This control does *not* show two things. Dropping the suspect does not leave the zeroed set unchanged — it costs it one member — and it does not raise the credit share: of the 9 drops only M raises it, and only M, G8 leave the set itself intact. What is robust is the floor, not the invariance. Likewise the minimiser is a subset of that lattice's own zeroed guards in 9 of 10 rebuilds rather than all of them; in the exception the minimiser scores +0.0040 on leave-one-out against +0.0920 for the guard ablation ranks first in the same lattice, a 23× separation. The effect is a property of composing released guards, not of any one of them, and it is a floor rather than an identity.

The harm-preserving deployment, read off the lattice. Every result above prices a benign–harmful trade-off through the scalar $v = H + \lambda F$, and the subset that minimises it at $\lambda = 1$, {W}, buys its advantage partly on the harmful side: it passes 0.0390 where the full stack passes 0.0000. A deployment team is usually not offered that trade, so we also report the lexicographic answer — hold H at the full stack's value and minimise F subject to it. Because the lattice is enumerated rather than sampled this is a lookup, not another experiment: 346 of the 512 coalitions hold H at exactly 0.0000, 4 of them tie at the lowest benign rate, and the smallest is the 2-guard {S,Q}, which blocks 45 of 250 benign prompts against the full stack's 0.5760 rate — 3.20× less over-refusal at no loss of harmful coverage. It is also an in-sample choice. On the other suites it passes harmful prompts on OR-Bench (1), and refitting the cheapest harm-free subset on half the main-suite clusters recovers {S,Q} in 90 of 200 splits and stays harm-free on the held-out half in 90: unlike the loss ratio, the harm constraint does not survive being held out, so treat the set as a candidate to validate. Note what it is not: only S of its members is in {L, S, L4, W, L1}, so the "the minimiser is drawn from the guards ablation zeroes" reading of §4.2 is a statement about the λ-weighted objective and does not carry over to the harm-constrained one. Both subsets are recoverable from Table 4 by a reader who distrusts our arithmetic.

Matching the guards to a common benign false-positive rate shrinks both pathologies several-fold on the XSTest-safe pool, and that is the paper's most actionable finding. The operating-point sweep of Appendix D moves all 8 guards to a common *threshold*, which preserves exactly the vendor calibration mismatch it claims to test: a threshold of 0.5 means different things to 8 differently-calibrated models. The control it should have run instead matches each guard to a common benign false-positive rate on the benign pool. Table 28 runs it at three targets. At a 3% per-guard benign FPR the full-stack loss falls from 0.576 to 0.148 while the best subset does not move at all — it is still {W}, at 0.059 against 0.051 at the vendor defaults — so the gap collapses to 2.50×; at 5% the gap is 2.76× and ablation zeroes no guard at all. Across all 3 targets the set ablation cannot see never exceeds 2 guards holding 16.4% — its worst case, at 3% — against 5 holding 71.5% at the defaults, a 4.4× reduction in credit mass at worst. It is a reduction, not an elimination: some guard still has a leave-one-out of exactly zero at 2 of the 3 targets. The direction matters and we want to be exact about it: the gap collapses because the stack gets better, not because the good subset gets worse. It also costs us the best-subset claim of §4.2 on this lattice, and we say so here rather than leave it to be found: at none of the three targets is the minimiser drawn from the guards leave-one-out zeroes. At the 3% target the zeroed set is {G, G8} while the minimiser is still {W}. Under matched calibration the exact-zero pathology of §4.2 shrinks by 4.4× in credit mass rather than disappearing. The control also moves the over-refusal rates of §3.1 and the saturation of the harmful side; the paragraph below reports how. What survives is the strongest form of the result we can honestly report — all three pathologies are properties of composing guards at their mismatched vendor defaults, and all three move when the guards are calibrated instead — and it makes the paper's advice two cheap things rather than an enumeration. Calibrate the guards to a matched benign FPR before composing them. Then do not stop at the ablation table: eliminating greedily to a stopping point cost 46 reads on this suite and was exact here.

What the same control does to the other two failures. Table 1, printed in §3.2 because that is where the saturation result is stated, reads the same three matched lattices for the two quantities the sentence above once put out of reach. Both move, and in the direction the paper wants. On the benign side the union's XSTest-safe block rate falls from 57.6% at the vendor defaults to 14.8% at the 3% target — 3.9× — with 20.8% and 34.8% at the two looser targets: the over-refusal headline of §3.1 is not untouched by calibration, it is the thing calibration most improves. On the harmful side the saturation partly lifts. At the vendor defaults no harmful prompt in any suite has a lone blocker (0 sole-blocked, the fewest blocking any single item being 3), so $H(N \setminus \{i\}) = H(N)$ for every guard and the harm half of every ablation difference at the vendor defaults is identically zero. Matched, 8, 4 and 1 harmful prompts have exactly one blocker at the three targets, which is exactly the quantity a single-component ablation reads. It buys that at one cost and not three: the minimum blocker count falls to 1, while the stack still passes 0 harmful prompts and $H(N)$ stays at 0.0000, exactly as at the vendor defaults. Recovering a sole blocker is a change in *who* is attributable, not a loss of coverage.

Two things follow, and the second is a scope restriction on a result this paper leads with. First, one recommendation acts on both failures we report, though it repairs neither fully. Matching the guards to a common benign FPR before composing them is simultaneously the cheapest reduction we found in the over-refusal rate, which stays far above the per-guard target, and the change that returns part of the safety signal to the instrument; we did not design it to do both and report the coincidence rather than claim it. Second, saturation is a property of how these 9 guards ship, not of the any-one-blocks composition rule. At the shipped operating points the harmful column is constant under every removal, which is why Appendix B's intervals are degenerate and why its multiplicity ceilings are ceilings; calibrated, none of that holds. A reader should attach "at the vendor defaults" to every saturation figure in this paper, and should not read it as a claim that composing safety guards by OR always destroys the ablation signal. These matched lattices are fitted in sample; with the cut points refitted out of sample, the held-out gap is 2.79 [2.06, 4.29]× against 2.50× in sample at a realised benign rate of 16.0% (Table 31).

Three caveats on this control, the first discharged by a held-out refit. The thresholds are fitted *in sample*: each guard's cut point is chosen on the same benign pool the resulting rate is then reported on, so every matched-FPR number here and in §5.1 is fitted rather than held out. Reading the in-sample fit as an upper bound on what matching buys is right for the *loss* and backwards for the *gap*, and we report both. Fitting the cut points on half the benign clusters and evaluating on the other half over 200 splits gives a gap of 2.79 [2.06, 4.29]× against the 2.50× fitted in sample — *larger* held out, so the recommendation survives being held out rather than depending on the fit. The loss levels behind

Table 28: Matching the guards to a common benign false-positive rate — the control the operating-point sweep should have run, since a common *threshold* preserves exactly the vendor calibration mismatch it claims to test. The upper panel is the per-guard threshold on p_{unsafe} at which each guard blocks at most the target share of the main suite's benign pool; S exposes no score and keeps its keyword decision, so it is not matched. The realised union rate is far above the per-guard target because it is the rate at which *any* of the 9 blocks, with S unmatched inside it, though still 3.9× below the 57.6% of the defaults. The lower panel is the resulting lattice. The pathology shrinks: at a 3% per-guard target the full-stack loss falls from 0.576 to 0.148 while the best subset does not move — the same coalition, at 0.059 against 0.051 — and at the two looser targets the gap is 2.76× and 3.74×. Across all 3 targets the set ablation cannot see never exceeds 2 guards holding 16.4% (its worst case, at 3%), against 5 holding 71.5% at the defaults — a 4.4× reduction in credit mass at worst, not an elimination. The collapse is the stack getting better, not the good subset getting worse. At none of the three targets is the minimiser drawn from the guards leave-one-out zeroes, which is the best-subset claim of §4.2 failing under this control. The over-refusal measurement and the saturation finding move with it too: Table 1 reads these same 3 lattices. Every threshold in the upper panel is fitted *in sample*: each guard's cut point is chosen on the same benign pool the resulting rate is then reported on, and every use of these numbers, in the body and in Appendix G alike, inherits that. That is not a best case: fitting the cut points on half the benign clusters and evaluating on the other half over 200 splits gives a gap of 2.79 [2.06, 4.29]× against the 2.50× printed here — *larger* held out, so the in-sample panel is the conservative one (Table 31).

target	G	L	S	M	G8	Q	L4	W	L1	realised (union)
3%	0.9831	0.5000	–	0.9579	0.9820	0.9670	0.9999	0.3208	0.9466	14.8%
5%	0.9740	0.1480	–	0.9325	0.9669	0.9176	0.9770	0.2018	0.7058	20.8%
10%	0.9241	0.0180	–	0.8933	0.8808	0.7076	0.0003	0.0421	0.2227	34.8%

resulting lattice at $\lambda = 1$

target	full-stack loss	best	gap	minimiser	zeroed by LOO
3%	0.148	0.059	2.50×	{W}	{G, G8}
5%	0.208	0.075	2.76×	{W}	none
10%	0.348	0.093	3.74×	{S}	{G8}

that ratio do move the other way: the full stack costs 0.160 held out against 0.148 in sample, so the in-sample collapse is a mild best case even though the ratio is not (Table 31). The other two caveats stand. S exposes no score, so it cannot be matched and enters every row at its keyword decision; and the realised union rate — the rate at which *any* of the 9 blocks a benign prompt — is far above the per-guard target at every setting, 14.8% in sample and 16.0% held out against a 3% target, because 8 matched guards and an unmatched S still compose by any-one-blocks. Matching the components does not by itself give the composition a 3% false-positive rate. It is 3.9× below the 57.6% the defaults produce, which is the largest single improvement any control in this paper buys and still not a solution to the composition rule.

Which benign pool the control is fitted on decides how large it looks. Every matched-FPR figure above fits the per-guard cut points on one benign pool, the main suite's XSTest-safe, and reports the effect on that same lattice. Refitting the identical control on each of the 4 benign benchmarks in turn (Table 29, evaluated on the main-suite lattice throughout) moves both headline effects several-fold: the union's benign block rate falls 2.57–7.20× and the credit ablation cannot see shrinks 1.7–4.4×, against 3.9× and 4.4× for the pool we report. The direction is the same everywhere and the size is not, so the calibrated numbers in this paper should be read as *the effect on XSTest-safe*, not as what calibration buys in general. One consequence is sharper than a spread: only 1 of the 4 pools leaves $H(N)$ at exactly zero. Fitted on the other three, the stack that was harm-free at the vendor defaults passes up to 8 harmful prompts — the benign gain is partly bought with harmful coverage, which the in-sample column alone cannot show. A team calibrating a stack should therefore fit on the benign distribution it actually serves, and re-measure the harmful side afterwards rather than assuming saturation survives.

Three out-of-sample and convention controls. Table 31 runs them; for each we say which weakness it retires and what it leaves standing. None of them re-runs a model either: all three are re-reads of the same stored decision matrices, split or reweighted.

Table 29: The matched-FPR control refitted on each benign benchmark and evaluated on the main-suite lattice, at the 3% target. The paper's row is XSTest-safe: its effect sizes are properties of that benign pool, not of calibration as such. Across the 4 pools the union's benign block rate falls 2.57–7.20×, the credit ablation cannot see shrinks 1.7–4.4×, and only 1 of them leaves $H(N)$ at zero — the others buy the lower benign rate with up to 8 harmful prompts passed. *pass* is harmful prompts the full stack lets through.

thresholds fitted on	n	union F	cut	$H(N)$	pass	gap	optimum	zeroed %
xstest-safe	250	0.148	3.89×	0.000	0	2.50×	{W}	16.4
jbb-benign	100	0.080	7.20×	0.016	8	1.23×	{L,S,Q,L1}	38.3
orbench-benign	1319	0.116	4.97×	0.008	4	1.60×	{L,S,Q,L1}	40.8
phtest-harmless	500	0.224	2.57×	0.002	1	2.77×	{L,S,W}	41.9

Held-out matched false-positive rate retires the in-sample caveat on the one recommendation this paper makes. The cut points are fitted on half the benign clusters and the resulting rate evaluated on the other half, over 200 splits: the gap is 2.79 [2.06, 4.29]× against 2.50× fitted in sample, at a realised held-out benign rate of 16.0% against 14.8% in sample. The levels behind that ratio are in the table too, 0.160 for the full stack against 0.059 for the best subset, because a ratio alone hides which side moved. The held-out figure is the larger, so the failure the caveat was written against did not occur and the in-sample number was the conservative report. What it does not retire is the second caveat above: the realised union rate is above the per-guard target held out exactly as it is in sample, because that is the composition rule and not the calibration.

Split-half selection retires the sharpest statistical objection available against the loss gap, that it is a ratio whose minimising subset is chosen on the same clusters the ratio is computed from. Selecting the argmin on half the clusters and evaluating its gap on the other half, over 2000 splits, gives 10.41 [5.18, 17.45] against 11.30 in sample: the winner's curse costs 7.9% of the gap and the held-out interval covers the in-sample value comfortably. The selection is stable enough to explain why so little is lost — the two halves choose the same subset in 60.7% of splits, and {W} is chosen in 80.6% of them.

The Owen value retires the objection that the conventions we printed were the ones convenient to us. The provenance partition, 5 guard-model vendor blocs plus singleton S, is a real coalition structure, and Owen's value for games with a priori unions (Owen, 1977) is the solution concept that respects it, averaging only over the orderings that keep a vendor's guards together. On this lattice it is efficient to machine precision (residual $< 10^{-12}$) and gives the zeroed set 63.9% of the credit under the $\sum_i |\phi_i|$ convention and 106.5% under the efficiency denominator: a sixth row of Table 20 and, like the other five, a different number. It also does what no rescaling of the Shapley value does, and this cuts against us: it *reorders* the guards, moving 6 of 9 without changing any sign — the per-guard values and the comparison against ϕ are given once, above. The set leave-one-out sends to zero is unchanged, because leave-one-out does not depend on the weighting, and that set is what this paper's claim is about. The reading is the same one §4.3 already takes from the other five conventions, now with a solution concept that moves the ordering as well as the share: the credit is not a stable quantity, and which guards ablation cannot see is.

The size phase is not same-vendor duplication. Three of the 9 sources are Meta checkpoints and two are IBM's, so the collapse in Table 2 could be a coalition filling up with relatives rather than with independent guards. Forcing one guard model per vendor, with S kept as a singleton player, answers it from the same stored decisions. Table 30 runs every one of the 6 representative choices. At $m = 3$ the harmful leave-one-out is already zero on 23.3–35.0% of (coalition, member) pairs, at $m = 4$ on 46.7–60.0%, and every roster has median H of zero at $m = 4$ with 99.1–99.2% of blocked harmful prompts carrying two or more blockers. The phase survives the deduplication.

Both at once: one guard per bloc at a matched 3% benign false-positive rate. The two de-pathologising moves above are applied separately; the most realistic form of the constructed-stack objection applies both. Re-thresholding every scored guard to the 3% target on the main-suite benign pool (S keeps its keyword decision) and re-reading the size phase on the same 6 rosters (`bloc_phase_fpr_v1.py`) weakens the phase without removing it. At $m = 3$ the harmful leave-

Table 30: The size phase of Table 2 after forcing one guard model per vendor, S kept as a singleton player: every representative choice, 6 rosters of 6. The collapse is not same-vendor duplication. At $m = 4$ every roster already has median H of zero, 99.1–99.2% of the harmful prompts a coalition blocks carry two or more blockers, and 46.7–60.0% of (coalition, member) pairs have a harmful leave-one-out of exactly zero, against 53.6% on the full roster. Re-read from the same stored decisions; no model is re-run.

roster	m=1	m=2	m=3	m=4	m=5	m=6
{W,Q,M,G,L,S}	0.0%	10.0%	33.3%	58.3%	86.7%	100.0%
{W,Q,M,G,L1,S}	0.0%	10.0%	33.3%	60.0%	80.0%	100.0%
{W,Q,M,G,L4,S}	0.0%	10.0%	35.0%	58.3%	80.0%	100.0%
{W,Q,M,G8,L,S}	0.0%	3.3%	23.3%	46.7%	80.0%	100.0%
{W,Q,M,G8,L1,S}	0.0%	0.0%	23.3%	55.0%	80.0%	100.0%
{W,Q,M,G8,L4,S}	0.0%	3.3%	25.0%	48.3%	73.3%	100.0%

Entries are the share of (coalition, member) pairs whose harmful leave-one-out is exactly zero.

Table 31: Three controls for calibration overfitting, subset-selection optimism and attribution-convention sensitivity. *Held-out matched FPR*: cut points fitted on half the benign clusters and the rate evaluated on the other half, so the recommendation is no longer fitted in sample; the held-out gap is if anything larger than the in-sample one, which makes the in-sample figure the conservative report. *Split-half selection*: the loss-minimising subset chosen on half the clusters and its gap evaluated on the other half, which prices the winner's curse on a ratio whose subset is selected on the same data; the median held-out gap is 7.9% below the in-sample one and {W} is selected in 80.6% of splits (the halves agree in 60.7%). *Owen*: the vendor partition is a real coalition structure, and the Owen value respects it; it is efficient here to machine precision, it reorders the guards against Shapley, and it leaves the set leave-one-out zeroes exactly where it was.

control	quantity	value
held-out matched FPR	gap, median over 200 splits	2.79 [2.06, 4.29]
	in-sample gap at the same target	2.50
	full-stack loss, held out	0.160
	best-subset loss, held out	0.059
	realised benign rate, held out	16.0%
split-half selection	held-out gap, 2000 splits	10.41 [5.18, 17.45]
	in-sample gap	11.30
	subset agrees across halves	60.7%
	most-selected subset	{W} (80.6%)
Owen (provenance blocs)	share of credit, $\sum_i \mid \cdot_i \mid$	63.9%
	share, efficiency denominator	106.5%
	efficiency residual	$< 10^{-12}$

one-out is zero on 6.7–13.3% of (coalition, member) pairs and at $m = 4$ on 20.0–28.3%, against 46.7–60.0% at the vendor defaults, with median H now 0.0058–0.0078 rather than zero; the full 6-guard rosters still read zero on 50.0–66.7% of pairs, with 98.2–98.4% of blocked harmful prompts carrying two or more blockers and median H of 0.0000–0.0019. Calibration plus deduplication moves the regime; it does not dissolve the redundancy.

Under the Banzhaf value the credit mass is much smaller. Table 32 puts the exact Shapley value beside the Banzhaf value on the same lattice at $\lambda = 1$. The two agree on the zeroed set, necessarily, since leave-one-out does not depend on the weighting; they disagree on how much credit that set carries, 71.5% against 4.4% — both under the $\sum_i |\phi_i|$ convention, two of the six rows of Table 20 — and on the raw sign disagreements, which fall from 1 to 0. The argument for permutation averaging in this particular decision is that a team removes guards in some order and the value should not depend on which order we imagine; the argument against is that Banzhaf's uniform weighting over coalitions is the closer match to a team that is choosing a subset rather than building one up. We do not think either argument settles it, so we report both and rest the comparison on what both agree about on this

Table 32: Solution-concept sensitivity on the vendor-default lattice at $\lambda = 1$: the exact Shapley value, the Banzhaf value, and leave-one-out for every guard, with the three lattice-level quantities beneath them. The two values agree on which guards ablation cannot see — the zeroed set is identical, since leave-one-out does not depend on the weighting — and disagree on how much credit that set carries: 71.5% under Shapley against 4.4% under Banzhaf, with the sign disagreements falling from 1 to 0. Banzhaf weights every coalition equally; Shapley weights every arrival order equally. We report both and do not treat the Shapley figure as the only reading; both shares are the $\sum_i |\phi_i|$ convention, one of six in Table 20. *dis.* is a *raw-sign* count — guards with $\phi_i < 0$ against a leave-one-out above 10^{-3} — because the cluster bootstrap exists only for the full-stack main-suite lattice and these rows have no intervals. On that lattice the single rule of §4.1, that a strict disagreement needs *both* intervals to exclude zero, gives 1 ({Q}) which at this roster size coincides with the raw count, leaving {G8} unsupported rather than contradicted. *share* is the $\sum_i |\phi_i|$ convention throughout; Table 20 gives that set under the other conventions, and a smaller one once S is held fixed.

Guard	ϕ_i (Shapley)	ϕ_i (Banzhaf)	LOO$_i$
G	+0.0840	+0.1754	+0.1080
L	-0.1022	-0.0047	+0.0000
S	-0.1052	-0.0044	+0.0000
M	+0.0464	+0.1193	+0.0560
G8	+0.0006	+0.0919	+0.0280
Q	-0.0646	+0.0355	+0.0160
L4	-0.0836	+0.0030	+0.0000
W	-0.1122	-0.0055	+0.0000
L1	-0.0871	+0.0018	+0.0000
zeroed by LOO			{L, S, L4, W, L1}
share of $\sum_i \|\phi_i\|$	71.5%	4.4%	
sign disagreements (raw)	1	0	

lattice: which guards leave-one-out cannot see, and that, at the vendor defaults, the loss-minimising coalition is drawn from them.

What the full stack is. We assembled it from publicly released checkpoints. It is not anyone's production system and we do not claim it as one. It is also not a good coalition: at $\lambda = 1$ exactly 1 of the 512 coalitions scores strictly worse — the empty one, which blocks nothing — and 31 more tie with it, so it sits at the bottom of the lattice rather than second from last. That is the point of the paper rather than an embarrassment: the union is deliberately untuned, and its role is to expose what full-stack removal measures under a common veto composition, the coalition-size, one-per-vendor, calibration and k-of-n controls testing how fast the regime fades as the construction is made less redundant (the crossed control above applies two of them at once). It does mean no result here is evidence about how a tuned production stack behaves.

H Artifact audit trail

Draft-level corrections that motivated several of the reporting rules above (the one-sided recapture certificate, the single rule for sign disagreements with exact zeros counted apart, the uniformly applied adjudicated corrections of Appendix C, the six credit conventions) are preserved with the released artifact in `ARTIFACT_AUDIT.md`, each with the earlier claim, the correction and the place in this paper where the rule now lives. No correction is omitted; every value printed in this version is regenerated from the stored decision records and checked by `check_numbers_v1.py`.