WHAT DOES BENCHMARK CAPABILITY COVERAGE ESTABLISH? AN AUDIT OF 16,217 BENCHMARKS

Anonymous authors
Paper under double-blind review

ABSTRACT

New benchmarks are often justified by a gap: no existing benchmark, it is claimed, tests some combination of capabilities. What can a map of benchmark coverage actually establish? We find that its answers depend on four choices that must be made explicit: how finely capabilities are distinguished, how many benchmarks must test a combination before it counts, whether a benchmark's labels are read as tested together, and which baseline defines "unusual". We make these choices explicit in a coverage-audit instrument whose logic is checked in Lean 4. Given the capability targets assigned to each benchmark, it reports which combinations are covered, which are rarer than the baseline predicts, and the benchmarks behind each answer. We apply it to 16,217 benchmarks labelled with 754 capabilities and measure the labels against two human raters. Coverage is concentrated beyond what benchmark breadth and capability popularity explain: a greedily chosen subset of 564 benchmarks covers every coarse capability pair the full corpus covers, against 812 in random corpora with the same breadths and popularities, and Papers-with-Code's own task tags show the same pattern. Which associations and modality restrictions look unusual changes with the baseline. The atlas also yields an exploratory list of under-covered pairs with the benchmarks behind them, and for one under-supported capability triple we show that an executable benchmark can be built and examined with facts-given and reduced-demand controls. Apparent gaps are targets for audit, not proof of absence. We release the vocabulary, labels, Lean development and pipeline.

1 INTRODUCTION

Benchmarks help decide which model improvements become visible (Dehghani et al., 2021), and new benchmarks are usually justified by a gap: no existing benchmark tests some capability, or some combination of capabilities. Before building one, researchers must determine whether the combination is already represented, under which conditions, and how strongly the evidence supports that answer. Whether such a gap exists is itself a measurement. Suppose a capability pair counts as covered once at least five benchmarks are labelled with both. Among the pairs assigned together to at least one benchmark in our corpus, 65.6 % are then covered when capabilities are named coarsely, but only 35.9 % when they are named finely (Figure 1). The same corpus supports opposite majority statements at the two resolutions. The majority also moves with the bar: fine coverage falls below half at a bar of three, while coarse coverage stays above half up to a bar of ten. We ask what a coverage claim over labelled benchmarks establishes, and how much of it follows from choices the map does not declare.

Existing maps of benchmarks and of the research literature count capabilities and their co-occurrences (Blagec et al., 2022; Ott et al., 2022; Fang et al., 2026). Four choices determine how such maps should be read, and they are rarely declared. The **resolution** fixes how finely capabilities are told apart. The **bar** fixes how many benchmarks must test a combination before it counts as measured. The **reading** fixes what a benchmark's labels assert together: several labels on one benchmark need not be demanded by the same item. The **reference model** fixes what counts as unusually frequent or rare. We make each choice explicit in a coverage-audit instrument: *given benchmark assignments, a vocabulary resolution and a reference model, it reports supported combinations, relative deficits and their witnessing benchmarks*. In plain terms, it says which combinations enough

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.

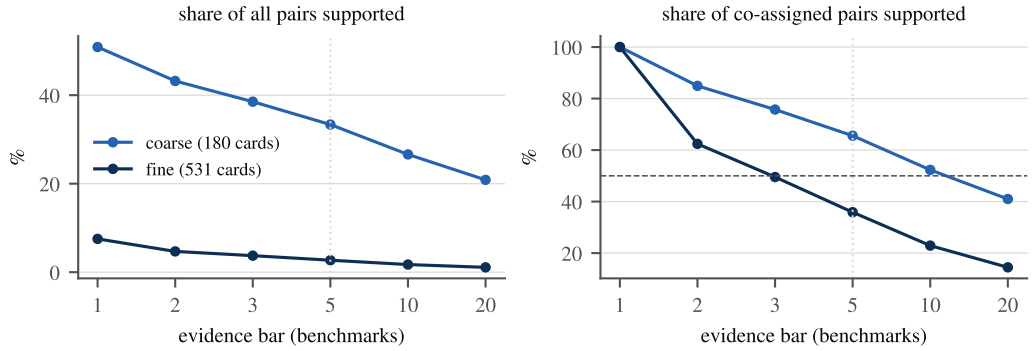


Figure 1: Whether most capability pairs count as covered depends on the bar and the resolution. At a bar of five, 65.6 % of co-assigned coarse pairs (5,374 of 8,196) are supported but only 35.9 % of fine pairs (3,796 of 10,584). Left: share of all pairs. Right: share of pairs co-assigned by at least one benchmark; dashed line at one half. Coarse: 180 cards; fine: 531. A coverage claim should therefore state its resolution and bar.

benchmarks test, which are rarer than the reference predicts, and which benchmarks each answer rests on. Its operational core is a benchmark-by-capability matrix; the added structure fixes how to aggregate it, condition it and read its uncertainty without changing what a resulting coverage claim means. A model checked in Lean 4 states what such a count establishes and what it does not (§2.1).

We instantiate the instrument as an atlas of 16,217 benchmarks and 754 capabilities (§2.2) and ask three questions of it. Is coverage concentrated beyond benchmark breadth and capability popularity? It is. Choosing benchmarks greedily, 564 suffice to cover every coarse pair the corpus covers; random corpora that keep every benchmark's breadth and every capability's popularity need 812. Papers-with-Code's own task tags agree (§3). Does the reference model change the answer? It does. A quarter to a third of the associations that look positive against independence do not exceed the reference by two standard deviations once each benchmark's breadth is also held fixed. The most frequent pair in Fang et al. (2026), for example, has lift 2.47 yet occurs less often than breadth and popularity predict. Likewise, testing a capability pair in only one input modality is common relative to a shuffle, but not once each capability keeps its own mix of modalities (§3). What can a practitioner do with the atlas? Before investing in a new benchmark, retrieve the existing benchmarks assigned the proposed capabilities, inspect those just below the threshold, and check whether the apparent under-coverage persists across resolutions, label policies and reference models. The result is an evidence-backed audit target; establishing that a demand is untested still requires reading task items. An exploratory queue of 115 under-covered pairs is one such output (§4), and one under-supported triple led to an executable evaluation, examined with facts-given and reduced-demand controls (§5).

Contributions.

- A reproducible coverage audit that reports supported capability combinations, relative deficits and supporting benchmarks under a declared resolution, bar, reading and reference, with conditional guarantees checked in Lean 4 (§2.1).
- Its instantiation: a typed vocabulary of 754 capabilities at four resolutions, 17 probe regimes and 14 setup dials, and labels for 16,217 benchmarks measured against two human raters (§2.2).
- An evaluation of what the instrument supports: concentration and reference-model dependence, a queue of inspectable deficit candidates, and one construction case (§3–5).

1.1 Related work

Evaluation science makes individual measurements more trustworthy: it standardises how a fixed set of benchmarks is run (Liang et al., 2023; Srivastava et al., 2023), audits benchmarks for quality and

validity (Reuel et al., 2024; Salaudeen et al., 2025) and calls for an evaluation science for generative AI (Weidinger et al., 2025). Catalogues map the ecosystem as a graph of tasks (Blagec et al., 2022; Ott et al., 2022), induce capability taxonomies (Wan et al., 2024; Wang et al., 2026), or map research attention over cognitive domains with lift against independence (Fang et al., 2026). Our object is a benchmark-level analogue of a Q-matrix (Tatsuoka, 1983), with no response table to check it against; Skill-Mix (Yu et al., 2024) shows that joint exercise can be established by construction but not inferred from co-occurring labels. Appendix C extends this discussion.

2 Instrument and data

2.1 What a coverage count establishes

A coverage claim is a statement about three things: a vocabulary of capabilities, a set of benchmarks, and a record of which capabilities each benchmark demands. This section defines each of them, and what a count over them establishes, with XNLI as a running example (Figure 4). Every definition and numbered result is a Lean 4 declaration checked against Mathlib; Appendix A states them in full, and Appendix B gives the proofs and the map to the Lean source. The checks cover the statements, not their application: a theorem fixes a direction or a bound under its hypotheses, and whether the atlas's counts meet those hypotheses (at bars above one, for pairs, or for machine labels whose sure and possible sets coincide) is argued in the text, not checked (Appendix B.8).

Definition 1 (Setting and corpus) *Fix a set of models $\mathcal{A}$, a set of benchmarks $\mathcal{B}$ and a finite set of cards $\mathcal{K}$ (Definition 2). Three maps connect them:*

- *the* answer key $\alpha : \mathcal{B} \to \mathrm{Ans}$ *gives each benchmark its correct answers;*
- *the* score $\sigma : \mathcal{A} \times \mathcal{B} \to \mathrm{Score}$ *gives each model its score on each benchmark;*
- *the* corpus $d : \mathcal{B} \to 2^{\mathcal{K}}$ *gives each benchmark its* demand $d(\boldsymbol{b})$*, the cards it is built to demand: a benchmark-level Q-matrix (Tatsuoka, 1983).*

More generally, a demand *is any set of cards $D \subseteq \mathcal{K}$, such as a combination whose coverage is in question. A* setting *is the pair (α, σ); the corpus is what labelling produces.*

Definition 2 (Cards) *The cards are $\mathcal{K} = \Phi \sqcup T \sqcup M$, of three kinds:*

- *a* capability *(Φ) is a property of models, identified with the set $\kappa \subseteq \mathcal{A}$ of models that have it;* $\boldsymbol{b}$ tests κ*, $\Theta(\boldsymbol{b}, \kappa)$, if some model with the property and some model without it receive different scores: $\sigma(\boldsymbol{a}, \boldsymbol{b}) \neq \sigma(\boldsymbol{a}', \boldsymbol{b})$ for some $\boldsymbol{a} \in \kappa$ and $\boldsymbol{a}' \notin \kappa$;*
- *a* probe regime *(T) is a map $\rho : \mathcal{B} \to \mathcal{B}$ with $\alpha \circ \rho = \alpha$: the same items presented differently;*
- *a* dial *(M) is a function $\mu : \mathcal{B} \to V$ of the benchmark alone (input modality, answer format).*

Φ and T carry a preorder $\leq$, "at least as specific as", and no order relation crosses kinds.

These are restricted formal abstractions. Θ does not establish that the property explains the score difference, and the operational vocabulary uses documented evaluative targets, broader probe contracts and explicit missing metadata (Appendix B.8). Benchmark labels describe what a benchmark is intended to require, not what it actually tests, which also depends on the models being compared (Proposition 6). Our coverage counts are therefore counts over intended *demands*.

Definition 3 (Demand order) *For demands $D, D' \subseteq \mathcal{K}$, D* demands at least *D', $D \sqsupseteq D'$, if every card of D' is met by a card of D at least as specific: $\forall c' \in D'\ \exists c \in D : c \leq c'$.*

This is the upper (Smyth) preorder on finite sets: demanding a more specific card, or more cards, is demanding at least as much. For example, our labelling (Section 2.2) assigns XNLI, among other cards, `multilingual` and `reasoning` (Figure 4), so

$$d(\text{XNLI}) \sqsupseteq \{\texttt{multilingual}, \texttt{reasoning}\} \sqsupseteq \{\texttt{generation}, \texttt{reasoning}\} \sqsupseteq \{\texttt{generation}\}.$$

The first step drops XNLI's other cards, the second generalises `multilingual` to its parent, and the third drops a card, so XNLI covers each of these demands (Definition 4). The converse of the second step fails: nothing in $\{\texttt{generation}, \texttt{reasoning}\}$ is as specific as `multilingual`.

Definition 4 (Coverage and gaps) *A corpus d* covers *a demand D, $\mathrm{Cov}_d(D)$, if $d(\boldsymbol{b}) \sqsupseteq D$ for some benchmark $\boldsymbol{b}$. Given the demands an analysis admits, an admitted demand that d does not cover is a* gap*.*

Cov reads a benchmark's cards as exercised jointly. This means the reported gaps are conservative, but not the coverage claims (Theorem 10). This holds given the assigned labels: a missed label makes a gap anti-conservative.

Definition 5 (Working cut and support) *A* working cut *is a finite antichain $W \subseteq \mathcal{K}$. The* projection $\pi_W(D) = \{w \in W : \exists c \in D,\ c \leq w\}$ *is the set of cut cards D settles. A cell $X \subseteq W$ has support* $\mathrm{supp}_d(X) = \{\boldsymbol{b} : X \subseteq \pi_W(d(\boldsymbol{b}))\}$ *and is* supported at bar *q if $|\mathrm{supp}_d(X)| \geq q$. Each card declares a level, and the cards of one level form a* rung*, which is a working cut.*

For example, `reasoning` and `generation` are umbrella cards and neither lies below the other, so $W = \{\texttt{reasoning}, \texttt{generation}\}$ is a working cut; $\{\texttt{multilingual}, \texttt{generation}\}$ is not, since `multilingual` lies below `generation`. Projecting the XNLI cards of Definition 3 gives $\pi_W(\{\texttt{multilingual}, \texttt{reasoning}\}) = \{\texttt{generation}, \texttt{reasoning}\}$: `multilingual` settles its parent, so XNLI supports the cell $\{\texttt{generation}, \texttt{reasoning}\}$ at W.

Closing a demand upward changes no projection. Coarsening only adds coverage and refining only reveals gaps, but the number of empty cells has no invariant value: one benchmark and two nested cuts can give zero gaps at the coarse cut and two at the fine one (Theorem 14). An uncovered card may still sit under a covered broader card, so a fine gap need not be a coarse one.

Uncertain labels. When an assignment is uncertain, each benchmark has a *sure* demand and a larger *possible* one (Definition 15). If the true demand lies within this envelope, every coverage count lies between the counts under the two, and neither bound can be tightened without more evidence (Theorem 16). The unanimous and the union merge of two annotators are these two ends only under that containment assumption.

Combinations and reference models. Under the joint reading of Definition 4, a benchmark covers every combination of the cut cards it settles, so the cells covered at order k are the union over benchmarks of the k-subsets of each projection, and together they form a simplicial complex. At the cut W above, XNLI contributes the order-2 cell $\{\texttt{generation}, \texttt{reasoning}\}$ and, with it, both of its order-1 cells. Under any weaker reading the covered cells can only shrink (Theorem 10). The complex's size is capped by the breadth of the benchmarks, and the Kruskal–Katona theorem caps each order by the one below (Theorem 18), so coverage at order $k+1$ is reported as a *fill* of that attainable cap. To separate concentration from these constraints we compare with random corpora of the same margins, read the same way: a Curveball trade exchanges cards between two benchmarks so that every benchmark keeps its breadth and every card its popularity exactly.

Conditioning. For a dial or probe-regime value v, conditioning keeps the benchmarks with that value and their demands unchanged. It never adds coverage. A combination remains undetermined when its known supporters share a value but another supporter is silent; known supporters with no common value already establish that it is not confined. A combination is covered only under v exactly when the benchmarks with v cover it and the others do not (Theorem 19).

2.2 The atlas

Corpus. We collect benchmark records from paper repositories, code hosts and dataset hubs and keep anything that can serve as a capability test: 16,217 records, from MMLU, GPQA and SWE-bench to an air-traffic-control speech suite and the 16PF personality questionnaire. The result is a mixed collection of machine-learning and scientific evaluation datasets, not of language-model benchmarks alone (10,713 records state no language). It is a collection, not a probability sample of the field.

Table 1: The labels against people and an external map. Gold: 100 benchmarks; intervals are benchmark bootstraps. Intersection gold counts a pair positive if both raters kept it, union gold if either did. These are performance statistics, not the bounds of Theorem 16. Details and further comparisons in Appendix E.

comparison	result
closed labels vs closed intersection gold, leave-one-benchmark-out	P 0.700, R 0.683, F_1 0.692 $[0.658, 0.721]$ (in sample 0.697)
closed labels vs closed union gold	P 0.813, R 0.529, F_1 0.641
human vs human, raw gold	κ 0.640 $[0.596, 0.680]$
external map (RepBench), fine cards	AUC 0.853 $[0.804, 0.901]$; recall 45–52 %
which pairs are holes, vs a human-standard model	hole-set Jaccard 0.37
human spot check: joint item found, covered vs hole pairs	14/16 vs 6/16, one-sided Fisher $p = 0.0046$

Vocabulary. The vocabulary has 754 evaluative facets at four rungs (16 umbrella, 180 coarse, 531 fine, 27 atom cards; Table 6), 17 probe regimes and 14 setup dials. Every card has a definition, positive and negative anchors and a contrastive example against its nearest neighbour. By origin, 179 facets are imported from existing frameworks, 110 are induced from the corpus and 460 are hypothesised, including 355 of the fine cards: drafted by a language model from a large mixed text corpus beyond the benchmark literature (research articles, cognitive science, education, fiction, economics and other fields). Three reviewers curated the whole vocabulary by hand, editing each card, rewriting parts the model drafted inadequately, and admitting it by majority vote. By status, 298 are marked speculative, 249 of them fine. Appendix D gives details.

Labels. A screening model (TypeSafe AI, 2026) scores all 12.2 million (benchmark, facet) pairs. The 1.36 million pairs at or above a gate of 0.11 are re-judged by two judge models (DeepSeek, 2026a; OpenAI, 2026), and a pair's score is half the screening score plus half the judges' mean. A facet is assigned at $\tau = 0.47$, the F_1-optimal threshold on the intersection gold, and assignments are closed upward. The run cost $267 (Appendix E).

Human gold. Two of the authors independently labelled 100 stratified benchmarks, 7,603 (benchmark, card) pairs in the closed comparison (7,598 raw rows scored for κ). Their intersection and union give two annotation policies. They would bound the true demand, as in Theorem 16, only if it lay between them, which our agreement measurements do not establish: both raters can miss a demand.

Resolution and bar. The fine majority holds only at a bar of two (62.4 %), the coarse majority up to a bar of ten (52.3 %; Figure 1). The coarse–fine difference at a bar of five is 0.297, with resampling sensitivity ranges of $[0.245, 0.284]$ (benchmark bootstrap) and $[0.282, 0.301]$ (80 % subsample); these are not confidence intervals, since duplicating benchmark rows changes witness multiplicities, and the observed difference need not lie inside them. The ordering holds in every resample and for label thresholds from 0.25 to 0.62, not at 0.67. Without speculative members the shares are 70.7 % and 42.8 %. The size of the difference is empirical: Theorem 14 fixes only the direction in which coverage of a given demand moves, and the two shares are over different pair sets.

2.3 What the labels can bear

Table 1 collects the evidence; coverage is computed on upward-closed labels, so the closed comparison leads.[1] The external map shares documentation with our labeller and gives only positive labels, so agreement with it is convergent evidence, not independent validation. Changing the labelling policy changes which pairs are holes, and how many: a model of the human labelling standard supports 11,962 of 16,044 non-linked coarse pairs at expected support five, the labels of record 5,323.

[1] Raw labels against raw intersection gold reach F_1 0.649. Fine audited joint positives exceed marginal independence by 1.34 $[1.11, 1.59]$, so calibrating single labels does not recover joint coverage. Aggregate coverage rankings of umbrellas and modalities are comparatively stable over nine label settings (minimum Spearman ρ 0.83 and 0.96); individual pair rankings are not.

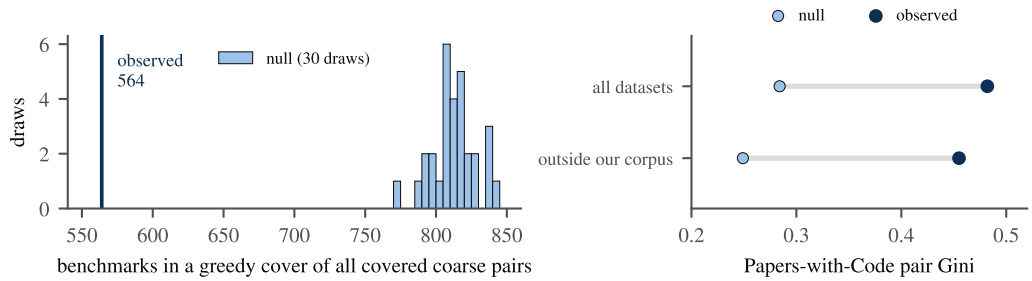


Figure 2: Coverage is concentrated beyond breadth and popularity. Left: benchmarks a greedy cover needs to witness every covered coarse pair, observed (564) and on 30 Curveball draws that keep every benchmark's breadth and every card's popularity (mean 812). Right: order-2 Gini of Papers-with-Code task-tag pairs, observed and under the same kind of null, for all datasets and for datasets outside our corpus.

Human spot check. One author judged 40 capability pairs from agent-prepared, verdict-free evidence sheets, blind to each pair's class but not to the project, asking whether some benchmark has single items that demand both capabilities (prespecified). A joint item was found for 14 of 16 covered pairs (0.875, $[0.640, 0.965]$) against 6 of 16 robust holes; the remaining 8 were policy-dependent low-support pairs, 5 of them with a joint item (descriptive). For 10 of the 16 holes (0.625, Wilson $[0.386, 0.815]$) none was found, which meets the prespecified point-estimate criterion but does not establish a rate above 0.60. A hole at a bar of five means fewer than five supporting benchmarks: rarely tested, not never. The check covers absolute-bar holes, not the relative deficits of §4. The labels do not establish the completeness that absence certification requires, so every gap below is conditional on them.

3 Coverage is concentrated, and the reference model decides what is unusual

This section answers the first two questions; §4 and §5 answer the third. All counts use the labels at $\tau = 0.47$, closed upward and read jointly. "Supported" means co-assigned by at least the bar; co-assignment is an upper reading of joint testing.

3.1 Concentration beyond breadth and popularity

A greedily chosen set of 564 benchmarks (3.5 %) witnesses every covered coarse pair (Figure 2). The same greedy procedure on 30 Curveball draws, which trade cards between benchmarks so that every benchmark keeps its breadth and every card its popularity exactly, needs 812 on average (95 % range 784–839; the observed value is below every draw). Greedy is not the minimum, and redundancy in labels is not redundancy in scores. Per covered pair the cover uses 0.069 benchmarks against 0.093 $[0.090, 0.097]$ under the null. It is a compact witness set for label coverage, not a replacement evaluation suite. In practice, a few percent of the corpus already witnesses every covered coarse pair; whether the rest is redundant is a question about scores, which labels cannot answer. The pattern holds outside our labeller: Papers-with-Code's human-curated task tags (10,175 datasets, 126 tasks) have an order-2 Gini of 0.482 against 0.284 under the same null, and 0.455 against 0.249 among datasets outside our corpus. This corroborates the concentration; it is not an independent replication, since the two collections are built from overlapping documentation.

3.2 The reference model decides which associations count

Fang et al. (2026) rank capability associations by lift against independence. On our matrix, through three crosswalks from their 14 domains to our cards, 24–33 % of the pairs with lift above one do not exceed the fixed-margin null mean by two standard deviations ($z < 2$), and 5–7 fall more than two standard deviations below it; no pair disagrees the other way. Take their most frequent pair,

Table 2: Three reference models and what each makes look unusual.

reference	holds fixed	a pair looks unusual when
independence (lift)	each card's frequency	it co-occurs more than the product of frequencies
fixed margins (Curveball)	every benchmark's breadth and every card's popularity	it co-occurs more or less than in random corpora with those margins
capability modality mix	the above, plus each capability's mix of input modalities	it is confined to one modality more than the capabilities' mixes imply

Language-Semantic $\times$ Reasoning. Against independence its lift, in their paper corpus, is 2.47: the two co-occur far more often than their separate frequencies predict. Against the fixed-margin null, which also keeps every benchmark's breadth, it falls below expectation in every arm. The first comparison licenses "more often than popularity alone predicts"; the second adds "but not more often than breadth and popularity together predict". Their highest-lift pairs keep their sign under both. The two baselines answer different questions, so the disagreement is not an error rate for their analysis (Table 2).

3.3 Setup is a reference too

If a capability pair is tested only on text, is that a restriction? Selecting benchmarks by a setup value, such as input modality, keeps a subset of benchmarks and never adds coverage (Theorem 19). Of 4,469 coarse pairs supported at a bar of five whose modality status can be decided, 289 are supported only by benchmarks that share one input modality (Table 13 in Appendix G). Shuffling modality values at random gives 9 such pairs; letting each capability keep its own mix of modalities gives 457. Single-modality pairs are thus far more common than a shuffle predicts, but fewer than the capabilities' own modality mixes predict (289 against 457, $z = -6.9$ over 10 draws). This is a comparison with a reference, not a causal decomposition. The count depends on how a parser default is read (1,135 when defaulted modalities count as values), and because modality is recorded per benchmark rather than per item, a single-modality pair is weaker evidence than a gap. Appendix G gives the remaining resolution and concentration statistics.

4 Deficit candidates are inspectable, not certified

A pair can be well tested in absolute terms and still be tested less often than its parts predict. For a pair of fine cards, let O be the number of benchmarks labelled with both, and E the number expected if benchmarks kept their breadth and cards their popularity (the bipartite configuration model, BiCM, which fixes both in expectation). A *deficit candidate* is a pair observed at most a quarter as often as expected, $O \leq E/4$, with $E \geq 10$, at each of three label thresholds, $\tau \in \{0.39, 0.47, 0.55\}$, refitting the model at each. Of 198 candidates that meet this rule both across the whole corpus and within each input modality, 115 (58 %) stay candidates under all three alternative labelling policies: a looser threshold fitted to the union gold, a model of the human labelling standard, and their conjunction with the labels of record.[2]

Table 3 shows the two policy-robust candidates with the largest within-strata shortfall $E - O$ and one illustration; the display was chosen after the policy results were seen. Both lead pairs also lie below an exact-margin null within the same strata. For each pair the atlas lists its witnesses and the benchmarks just below the cut: HLVU carries video understanding and scores 0.46 on relational reasoning, just under the 0.47 cut. Such candidates are possible missed labels as much as possible extensions. The queue is exploratory: it is selected by an effect-size rule, carries no p-values, is not

[2]Eligible: 147 non-speculative fine cards carried by at least 20 benchmarks, 10,731 pairs not linked by the ladder. The global rule gives 512 candidates and the within-modality rule 274, in parallel rather than in sequence; benchmarks whose modality is only a parser default form their own stratum. Thirteen of the 115 involve non-English script or multilingual cards; since monolingual exams can carry these cards at the same score as multilingual ones, they may be labelling artefacts. Appendix H gives the sets and the policy table; the supplement lists all 115 with their witnesses (`deficit_queue/deficit_queue_115.csv`).

Table 3: Named relative deficits (fine rung). *Observed*: benchmarks co-assigned both cards at $\tau = 0.47$ (at 0.39 / 0.55 in brackets). E: expected under the BiCM, fitted within modality strata and globally. The first two pairs are deficits under every labelling policy and lie below an exact-margin null within the same strata (Curveball, 200 draws, $z = -16.2$ and -13.9); the illustration is a deficit under the labels of record but not under the looser threshold. Cards: `graph-structured-reasoning-and-relational-inference`, `code-generation`, `factual-knowledge`, each with `temporal-visual-understanding`. Exploratory: selected by an effect-size rule, no p-values, not verified by independent reviewers; the human spot check of Section 2.3 covers absolute-bar holes, not this queue.

pair	observed	E stratified	E global
graph or relational reasoning × video understanding	12 (21 / 6)	158	112
code generation × video understanding	13 (34 / 8)	144	260
illustration: factual knowledge × video understanding	28 (80 / 11)	181	427

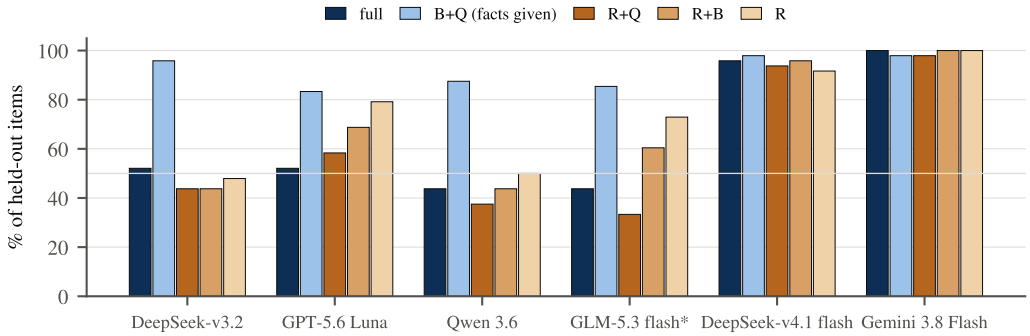


Figure 3: Construction case: share of the 48 held-out items solved per model, on the full question and its demand-removal controls. B: belief-updating, Q: quantitative reasoning, R: retrieval-in-context; B+Q is the facts-given pair. *GLM-5.3 flash hit a 32k output cap; capped responses count as errors. Counts in Table 14.

verified by independent review, and $E - O$ is not a number of benchmarks the field should build. It is robust to the labelling policy, not to how records with a defaulted modality are stratified: by our registered rule the within-modality set (274 here, 335 with those records counted as text) fails that check.

5 An under-supported cell becomes an executable evaluation

Can an atlas output be turned into a working test? From the atlas's coarse minimal non-faces at a bar of five (triples whose pairs are supported while the triple is not) we screened the top 80 by hand for text-only cells with an executable key and picked rank 67: `belief-updating` + `quantitative-and-mathematical-reasoning` + `retrieval-in-context`. The triple has four labelled carriers, so it is under-supported, not absent. Read against the demand our construction implements, revising a probability from evidence in a supplied long context, three carriers lack it and one is partial; this narrower reading neither invalidates their broader labels nor excludes carriers the labels missed. A hidden world fixes an exact posterior; a language model renders it as a 7.8–14.2k-word case file that a second model reads back, and every item comes with demand-removal controls. We evaluate six models on 48 held-out items (Appendix I).

Four models solve 44–52 % of full held-out items and 83–96 % when the relevant facts are given; this facts-given contrast holds for each of the four separately (Appendix I). The other two models solve 96–100 % and stay near ceiling on every control. We detected no additional full-item difficulty relative to either retrieval-containing pair control.

The result points to difficulty in extracting and using evidence from long prose for some models. It does not isolate retrieval from evidence tracking, distraction or other burdens of a long context, and it does not establish a three-way interaction; the two near-ceiling models show that an under-supported combination need not be an unsolved one. This is one feasible construction for one manually selected cell, not evidence that the atlas selects gaps better than an expert or a simple baseline; retrospectively, atlas-chosen pairs did not beat a pair-gap baseline at predicting which combinations were built later (Appendix J).

6 Discussion, limitations and conclusion

We have examined what a coverage map of 16,217 benchmarks establishes once its resolution, bar, reading and reference model are declared. The answer moves with each choice, the concentration of coverage survives controls for breadth and popularity, and a queue of exploratory deficit candidates remains to be inspected; separately, one under-supported triple found by the same atlas became a working benchmark.

The atlas turns a proposed coverage gap into an inspectable question: what evidence supports it, which choices change it, and which existing benchmarks should be examined before collecting more evaluation data. It is an audit of coverage conclusions, not an engine for discovering missing capabilities; we have not shown that it selects benchmarks, diagnoses models or saves evaluation effort better than simpler tools. The guidance is conditional. To report that a combination is rarely tested, state its resolution, bar and reference, and show its witnesses. A claim of zero coverage additionally requires a complete upper envelope of labels and an explicitly bounded corpus; a hole at a bar of five cannot establish that a combination is untested. To find where to look, use the deficit queue with its witnesses; to build, §5 shows one path from an under-supported cell to a benchmark. More generally, the capability cards, probe regimes and dials span a morphological (Zwicky) box (Zwicky, 1969) in which every cell is a candidate benchmark design, and minimal non-faces pick cells whose parts are tested while their combination is under-supported; Cell 67 is one cell drawn this way, and we have not evaluated the box as a generator.

What releases should contain. Everything above is recovered from documentation, which makes a sixteen-thousand-benchmark audit possible and is also its binding constraint. We believe coverage is best settled where the evidence is: in releases that declare, per item, which capabilities it demands and whether jointly, next to a long-format response table (Appendix L). Labels read from descriptions are fuzzy where a demand is not clear-cut: our raters agree at κ 0.640, and such cases admit several readings depending on the labeller. Item-level responses would let demands be estimated under a response model rather than read (Liu et al., 2012; de la Torre, 2009), and could expose dimensions that no card names. This matters because our vocabulary is drawn largely from literature on human abilities; the hypothesised cards anticipate where model capabilities may go, but we expect capabilities without a human analogue to surface in responses rather than in descriptions.

Limitations. The labels are approximate (F_1 0.692 leave-one-benchmark-out; 0.597 raw on the 75 gold benchmarks no design choice used), and the gold rests on two authors who agree at κ 0.640; label errors may be correlated across pairs. The vocabulary is ours: 460 of 754 capabilities are hypothesised, so empty cells partly reflect what we chose to name. More generally, structure in the atlas partly reflects our vocabulary and documentation-derived labels; the concentration result is the one we corroborate with labels we did not produce. Co-assignment is an upper reading of joint exercise. The corpus is a collection, not a sample. The BiCM reference fixes margins in expectation, the Curveball null exactly. The deficit queue is unverified by independent task review, and we found no demonstrated benefit for model diagnosis or selection (Appendix J). A prospective test across several cells (queue against a pair-gap baseline and random cells, with demand-removal controls), and treating language as a setting rather than a capability, are left for future work. Coverage maps remain useful instruments for evaluation design, but it is important to know which of their gaps belong to the benchmarks and which to the map.

AI USE STATEMENT

Language models are part of the method and are disclosed where they are used. A screening model and two judge models assign the capability labels (§2.2, Appendix E); their outputs are measured against two human raters and never replace the human gold. In the construction case, a language model renders items from structured worlds, another validates them by read-back, and six models are evaluated (Appendix I). The evidence sheets for the human spot check were prepared by language-model agents without verdicts; every verdict is a human one. We also used AI coding and writing assistants for pipeline code, analysis scripts and editing of the text. A language model drafted the hypothesised capability cards from a large text corpus; three human reviewers curated the whole vocabulary by hand, editing each card, writing by hand the parts the model drafted inadequately, and admitting it by majority vote. Released scripts recompute every number except the manual steps listed in Appendix I, the formal claims are checked in Lean 4, and the authors reviewed all AI-assisted work and take responsibility for the final content, including text, claims and artifacts.

ETHICS STATEMENT

The corpus consists of public benchmark records; we release capability labels and metadata, not the item text of the benchmarks we audit. The only human judgements are by the authors themselves (the gold labels and the spot check); no external participants were involved. Coverage gaps are reported as audit targets, not as certified absences, so that they are not read as evidence that a capability is untested.

REPRODUCIBILITY STATEMENT

We release the vocabulary, the labels of record, the human gold, the Lean development and the pipeline; Appendix K lists them. Every statistic and figure is recomputed from these released inputs by scripts that check their output against the printed values, and the formal claims build with Lean 4 and Mathlib. Recomputing statistics does not regenerate the labels: they come from stochastic language-model calls and are released as data, with their prompts and costs. The Cell 67 package recomputes that section except its two manual steps (Appendix I).

REFERENCES

Lorin W. Anderson and David R. Krathwohl (eds.). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Longman, New York, 2001.

Sanjeev Arora and Anirudh Goyal. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*, 2023.

John B. Biggs and Kevin F. Collis. *Evaluating the Quality of Learning: The SOLO Taxonomy*. Academic Press, New York, 1982.

Kathrin Blagec, Georg Dorffner, Milad Moradi, and Matthias Samwald. A critical analysis of metrics used for measuring progress in artificial intelligence. *arXiv preprint arXiv:2008.02577*, 2020.

Kathrin Blagec, Adriano Barbosa-Silva, Simon Ott, and Matthias Samwald. A curated, ontology-based, large-scale knowledge graph of artificial intelligence tasks and benchmarks. *Scientific Data*, 9(1):322, 2022. doi: 10.1038/s41597-022-01435-x.

Maty Bohacek, Nino Scherrer, Nicholas Dufour, Thomas Leung, Christoph Bregler, and Stephanie C. Y. Chan. Uncovering competency gaps in large language models and their benchmarks. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2512.20638.

Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, and Baishakhi Ray. Benchmarking large language models under data contamination: A survey from static to dynamic evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10080–10098, 2025. doi: 10.18653/v1/2025.emnlp-main.511.

Tsong Y. Chen, Shing C. Cheung, and Siu Ming Yiu. Metamorphic testing: A new approach for generating next test cases. Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, 1998.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Jimmy de la Torre. DINA model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, 34(1):115–130, 2009. doi: 10.3102/1076998607309474.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, pp. 625–635, 2021. doi: 10.1007/978-3-030-79876-5_37.

DeepSeek. DeepSeek-V4.1-Flash. Language model `deepseek/deepseek-v4.1-flash`, served by CoreWeave, 2026a. Accessed September 2026.

DeepSeek. DeepSeek-V4-Flash. Language model `deepseek/deepseek-v4-flash`, 2026b. Accessed September 2026.

Mostafa Dehghani, Yi Tay, Alexey A. Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals. The benchmark lottery. *arXiv preprint arXiv:2107.07002*, 2021.

Shixin Fang, Jiachen Wo, Wenjuan Qin, Sihang Jiang, and Yanghua Xiao. From isolated tasks to structured capabilities: A multilayer taxonomy for large language models. *arXiv preprint arXiv:2607.22182*, 2026.

Bradley Fowler, Ryan Smith, Daniel Thi Graviet, William Myers, Joshua Greaves, Narmeen Fatimah Oozeer, Antía García, Philip Quirke, Amirali Abdullah, Fazl Barez, and Shriyash Kaustubh Upadhyay. The capability frontier: Benchmarks miss 82% of model performance. *arXiv preprint arXiv:2606.26836*, 2026.

H. Paul Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan (eds.), *Syntax and Semantics, Volume 3: Speech Acts*, pp. 41–58. Academic Press, New York, 1975.

Anson Ho, Jean-Stanislas Denain, David Atanasov, Samuel Albanie, and Rohin Shah. A Rosetta stone for AI benchmarks. *arXiv preprint arXiv:2512.00193*, 2025.

Gyula O. H. Katona. A theorem of finite sets. In P. Erdős and G. Katona (eds.), *Theory of Graphs*, pp. 187–207. Akadémiai Kiadó, 1968.

Ryan Othniel Kearns. Quantifying construct validity in large language model evaluations. *arXiv preprint arXiv:2602.15532*, 2026.

Joseph B. Kruskal. The number of simplices in a complex. In Richard Bellman (ed.), *Mathematical Optimization Techniques*, pp. 251–278. University of California Press, 1963. doi: 10.1525/9780520319875-014.

John P. Lalor, Pedro Rodriguez, João Sedoc, and José Hernández-Orallo. Item response theory for natural language processing. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, pp. 9–13, 2024.

Jacqueline P. Leighton, Mark J. Gierl, and Stephen M. Hunka. The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka's rule-space approach. *Journal of Educational Measurement*, 41(3):205–237, 2004. doi: 10.1111/j.1745-3984.2004.tb01163.x.

Qingquan Li, Shaoyu Dou, Kailai Shao, Chao Chen, and Haixiang Hu. Evaluating scoring bias in LLM-as-a-judge. In *Database Systems for Advanced Applications (DASFAA 2026)*, volume 16538 of *Lecture Notes in Computer Science*. Springer, 2026. doi: 10.1007/978-981-92-0372-7_2. arXiv:2506.22316.

Xiang Lisa Li, Farzaan Kaiyom, Evan Zheran Liu, Yifan Mai, Percy Liang, and Tatsunori Hashimoto. AutoBencher: Towards declarative benchmark construction. In *International Conference on Learning Representations (ICLR)*, 2025.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. arXiv:2211.09110.

Jingchen Liu, Gongjun Xu, and Zhiliang Ying. Data-driven learning of Q-matrix. *Applied Psychological Measurement*, 36(7):548–564, 2012. doi: 10.1177/0146621612456591.

László Lovász. *Combinatorial Problems and Exercises*. North-Holland, 1979.

Kevin S. McGrew. CHC theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research. *Intelligence*, 37(1):1–10, 2009. doi: 10.1016/j.intell.2008.08.004.

Robert J. Mislevy, Linda S. Steinberg, and Russell G. Almond. On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives*, 1(1):3–62, 2003. doi: 10.1207/s15366359mea0101_02.

National Institute of Standards and Technology. Artificial intelligence risk management framework (AI RMF 1.0). Technical Report NIST AI 100-1, National Institute of Standards and Technology, 2023.

Shiwen Ni, Guhong Chen, Shuaimin Li, Xuanang Chen, Siyi Li, Bingli Wang, Qiyao Wang, Xingjian Wang, Yifan Zhang, Liyang Fan, Chengming Li, Ruifeng Xu, Le Sun, and Min Yang. A survey on large language model benchmarks. *arXiv preprint arXiv:2508.15361*, 2025.

OpenAI. GPT-6 Luna. Language model `openai/gpt-6-luna`, OpenAI Batch API, 2026. Accessed September 2026.

Simon Ott, Adriano Barbosa-Silva, Kathrin Blagec, Jan Brauner, and Matthias Samwald. Mapping global dynamics of benchmark creation and saturation in artificial intelligence. *Nature Communications*, 13:6793, 2022. doi: 10.1038/s41467-022-34591-0.

Norman G. Peterson, Michael D. Mumford, Walter C. Borman, P. Richard Jeanneret, Edwin A. Fleishman, Kerry Y. Levin, Michael A. Campion, Melinda S. Mayfield, Frederick P. Morgeson, Kenneth Pearlman, Marilyn K. Gowing, Anita R. Lancaster, Marilyn B. Silver, and Donna M. Dye. Understanding work using the occupational information network (O*NET): Implications for practice and research. *Personnel Psychology*, 54(2):451–492, 2001. doi: 10.1111/j.1744-6570.2001.tb00100.x.

Anka Reuel, Amelia Hardy, Chandler Smith, Max Lamparth, Malcolm Hardy, and Mykel J. Kochenderfer. BetterBench: Assessing AI benchmarks, uncovering issues, and establishing best practices. In *Advances in Neural Information Processing Systems 37, Datasets and Benchmarks Track*, pp. 21763–21813, 2024. doi: 10.52202/079017-0685. arXiv:2411.12990.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, 2020.

Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. doi: 10.1093/biomet/63.3.581.

Olawale Salaudeen, Anka Reuel, Ahmed Ahmed, Suhana Bedi, Zachary Robertson, Sudharsan Sundar, Ben Domingue, Angelina Wang, and Sanmi Koyejo. Measurement to meaning: A validity-centered framework for AI evaluation. *arXiv preprint arXiv:2505.10573*, 2025.

Sergio Segura, Gordon Fraser, Ana B. Sanchez, and Antonio Ruiz-Cortés. A survey on metamorphic testing. *IEEE Transactions on Software Engineering*, 42(9):805–824, 2016. doi: 10.1109/TSE.2016.2532875.

Elizabeth S. Spelke and Katherine D. Kinzler. Core knowledge. *Developmental Science*, 10(1):89–96, 2007. doi: 10.1111/j.1467-7687.2007.00569.x.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. arXiv:2206.04615.

Giovanni Strona, Domenico Nappo, Francesco Boccacci, Simone Fattorini, and Jesus San-Miguel-Ayanz. A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. *Nature Communications*, 5:4114, 2014. doi: 10.1038/ncomms5114.

Kikumi K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354, 1983. doi: 10.1111/j.1745-3984.1983.tb00212.x.

The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pp. 367–381, 2020. doi: 10.1145/3372885.3373824.

TypeSafe AI. Jev 1.13. Language model `typesafe/jev-1.13` (snapshot `jev-1.13-20260917`), accessed through the OpenRouter Decisions API, 2026. Accessed September 2026.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W. White, Longqi Yang, Reid Andersen, Georg Buscher, Dhruv Joshi, and Nagu Rangan. TnT-LLM: Text mining at scale with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5836–5847, 2024. doi: 10.1145/3637528.3671647.

Chao Wang. What do we expect from LLMs? mapping the design of LLM benchmarks. *arXiv preprint arXiv:2609.19182*, 2026.

Zongqi Wang, Tianle Gu, Chen Gong, Xin Tian, Siqi Bao, and Yujiu Yang. SCAN: Structured capability assessment and navigation for LLMs. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026. arXiv:2505.06698.

Yifan Wei, Li Du, Xiaoyan Yu, Yang Feng, and Angsheng Li. Towards compositional generalization of LLMs via skill taxonomy guided data synthesis. *arXiv preprint arXiv:2601.03676*, 2026.

Laura Weidinger, Inioluwa Deborah Raji, Hanna Wallach, Margaret Mitchell, Angelina Wang, Olawale Salaudeen, Rishi Bommasani, Deep Ganguli, Sanmi Koyejo, and William Isaac. Toward an evaluation science for generative AI systems. *The Bridge*, 2025. arXiv:2503.05336.

Dingli Yu, Simran Kaur, Arushi Gupta, Jonah Brown-Cohen, Anirudh Goyal, and Sanjeev Arora. Skill-Mix: A flexible and expandable family of evaluations for AI models. In *International Conference on Learning Representations*, 2024. arXiv:2310.17567.

Z.ai. GLM-5.3-Flash. Language model `z-ai/glm-5.3-flash`, 2026. Accessed September 2026.

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, Datasets and Benchmarks Track*, 2023.

Fritz Zwicky. *Discovery, Invention, Research through the Morphological Approach*. Macmillan, New York, 1969.

A THE FORMAL MODEL IN FULL

This appendix states in full the model that Section 2.1 summarises and every count in the paper is taken over. It fixes what a label asserts, what a benchmark is credited with covering, at what level of detail coverage is counted, what the labels leave undetermined, and how a count is conditioned on the conditions a test is run under. Every definition and every statement below is a Lean 4 declaration checked against Mathlib (de Moura & Ullrich, 2021; The mathlib Community, 2020); proofs, the remaining definitions and the correspondence to the Lean source are in Appendix B.

A.1 CARDS, AND THE THREE KINDS OF THING A LABEL CAN NAME

Capabilities are properties of the model, dials of the test, probe regimes of how it is presented. Fix a set of models $\mathcal{A}$, a set of benchmarks $\mathcal{B}$, an answer key $\alpha : \mathcal{B} \to \mathrm{Ans}$ and a score $\sigma : \mathcal{A} \times \mathcal{B} \to \mathrm{Score}$. A label can name three different types of object:

- a *dial* is a function $\mu : \mathcal{B} \to V$ of the benchmark alone — its input modality, answer format, scoring protocol;
- a *probe regime* is a map $\rho : \mathcal{B} \to \mathcal{B}$ with $\alpha \circ \rho = \alpha$ — the same items presented differently (paraphrased, translated, padded with distractors, put under pressure) without changing their correct answer. Maps of this type contain the identity and are closed under composition, so they form a monoid acting on $\mathcal{B}$ (a property of the abstraction, not of any catalogue of regimes); they need not be invertible, since padding with distractors cannot in general be undone (`Perturbation.exists_not_invertible`);
- a *capability* $\kappa \subseteq \mathcal{A}$ is a property of the model, and a benchmark tests it when its score separates a model that has it from one that does not:

$$\Theta(\boldsymbol{b}, \kappa) \;:\Longleftrightarrow\; \exists\, \boldsymbol{a}, \boldsymbol{a}' \in \mathcal{A} :\; \boldsymbol{a} \in \kappa,\; \boldsymbol{a}' \notin \kappa,\; \sigma(\boldsymbol{a}, \boldsymbol{b}) \neq \sigma(\boldsymbol{a}', \boldsymbol{b}).$$

A probe regime preserves the answer key, but not the score: it may change what a model scores on a benchmark, and it may destroy a test outright. The type is only the deterministic, answer-invariant case. Regimes that change the correct outcome, stochastic ones (resampling collapses to the identity) and adaptive ones fall outside it; among the 17 live tags these are `value-substitution`, `locale-evaluation-dependency`, `information-ablation`, `decomposition-variation` and `directional-alignment-test`. For them, `ProbeApplication` formalizes the structure of an application (a task contract as a parameter, a domain, an instance-dependent outcome relation and a declared assumption) and the obligations a valid one must discharge. It records, it does not validate: any map is an application with a vacuous relation (`vacuous_application`), so claims of preserved meaning need `ProbeApplication.NonVacuous` and claims resting on the assumption need `ProbeApplication.AssumptionSatisfied`. The answer key and probe regimes act on items; coverage is counted over benchmarks, whose demand is the union of their items' demands (`ItemCorpus.toBench`).

Proposition 6 (Testing is not fixed by the artefact) *There are two settings with the same benchmarks and the same answer key, a benchmark $\boldsymbol{b}$ and a capability κ such that $\boldsymbol{b}$ tests κ in one setting and not in the other. So no function of the benchmark artefact alone decides Θ in every setting.*

This is why our labels, which are read from documentation and are therefore functions of the artefact, record what a benchmark is documented to *target* — its demand, in the model — rather than what it tests. The proposition compares settings; it does not say that within one fixed setting no such function exists.

Proposition 7 (The tested capability is not fixed by the scores) *There is a setting and two capabilities κ, κ' that disagree on every model yet are tested by exactly the same benchmarks.*

So which benchmarks separate models does not, on its own, identify the capability they test: the vocabulary has to be supplied rather than read off score separation. Richer evidence or further assumptions may identify it; the proposition rules out only the separation pattern as the sole source.

The card space. The cards form a disjoint union $\mathcal{K} = \Phi \sqcup T \sqcup M$ of capability, probe-regime and dial cards. Φ and T each carry a preorder $\leq$, read $c \leq c'$ as "c is at least as specific as c'" (`deductive-reasoning` $\leq$ `reasoning`); M carries the discrete order; and cards of different kinds are never comparable. Every result below is stated for an arbitrary preorder and so holds on each axis, and on the whole of $\mathcal{K}$. (In the released taxonomy T is flat — its order is discrete too — which the model allows.)

The definitions, and the part of each that the types certify. The vocabulary defines its three kinds of card in prose; each type above is the part of a definition the Lean model states, and nothing more.

- *Evaluative facet* (Φ). A competence or behavioral disposition named as a target of evaluation. Assigning a facet to an evaluation asserts, on cited evidence and under a stated labelling rule, that the evaluation targets it. Formally, a facet names a capability κ, and the facets assigned to $\boldsymbol{b}$ form its demand $d(\boldsymbol{b})$. The model does not encode the evidence, the labelling rule, or whether a target is reported or supported; and Θ is score separation only, establishing neither demand nor identification.
- *Probe regime* (T). A specified procedure for varying the presentation or probing of task instances relative to a reference, within a declared task contract. The procedure preserves the task's rules and evaluation objective, and has a justified relation between reference and resulting correct outcomes on its stated domain of application. Formally, a probe regime is the invariance case: a total, deterministic, answer-preserving map ρ. Task mechanics, application domains, adaptive protocols and general outcome relations (a directional test, a substitution that maps the answer) are not encoded, and neither is the composition of two applications valid on different domains.
- *Metadata dial* (M). A named descriptor of the evaluation setup with a value domain and a reproducible extraction or measurement rule that does not infer the evaluative target. Its value records a specified property of that setup; it does not assert that an evaluative facet is targeted, required, possessed, or successfully measured, or certify that a probe satisfies its T contract. Formally, a dial is a function μ of the benchmark; the type does not enforce the extraction contract — for a fixed setting and capability, $\Theta(\cdot, \kappa)$ itself has the type of a dial.

In one sentence: ϕ identifies the evaluative target; T identifies a controlled probe variation; M records properties of the evaluation setup. Neither the three-kind partition nor its usefulness is a Lean theorem: the order only guarantees that no subsumption crosses kinds.

A.2 Demands and coverage

A benchmark exercises several cards at once; the finite set it exercises is its *demand* $D \subseteq \mathcal{K}$. A *corpus* is a map $d : \mathcal{B} \to 2^{\mathcal{K}}$ assigning each benchmark its demand — the Q-matrix of cognitive diagnosis (Tatsuoka, 1983; Liu et al., 2012), with benchmarks as items.

Definition 8 (Demand order) *D demands at least D', written $D \sqsupseteq D'$, iff every card of D' is met by a card of D at least as specific: $\forall c' \in D' \; \exists c \in D : \; c \leq c'$.*

This is the upper (Smyth) preorder on finite sets. It is the single relation behind both ways in which a benchmark exercises more than it names: dropping a conjunct ($D' \subseteq D \Rightarrow D \sqsupseteq D'$) and generalising along the ladder ($c \leq c' \Rightarrow \{c\} \sqsupseteq \{c'\}$). On a discrete axis it is exactly reverse inclusion, so dials and the flat probe regimes need no separate mechanics.

Definition 9 (Coverage and gaps) *A corpus d* covers *a demand D if some benchmark demands at least it, $\mathrm{Cov}_d(D) :\Leftrightarrow \exists \boldsymbol{b} \in \mathcal{B} : \; d(\boldsymbol{b}) \sqsupseteq D$; we drop the subscript when the corpus is fixed. Given an admissibility predicate adm on demands, D is a* gap, $\mathrm{Gap}(D)$, *if $\mathrm{adm}(D)$ and $\neg\mathrm{Cov}(D)$.*

The admissibility predicate excludes combinations that cannot exist, so that an impossible cell is not counted as a missing one; it is a parameter of the model. Coverage is downward closed ($\mathrm{Cov}(D)$ and $D \sqsupseteq D'$ imply $\mathrm{Cov}(D')$), so gaps are upward closed: an admissible demand stronger than a gap is a gap.

The joint reading is the conservative one for gaps. Cov reads the cards of a benchmark as exercised *jointly*. The corpus does not record whether they are: a conjunctive or a compensatory response model would credit different combinations. Model a reading as any relation $\boldsymbol{b} \vdash D$ ("$\boldsymbol{b}$ is credited with settling D") and say it *credits no more than the joint reading* if $\boldsymbol{b} \vdash D$ implies $d(\boldsymbol{b}) \sqsupseteq D$ — a benchmark cannot settle a demand whose cards it never names.

Theorem 10 (Gaps survive every combination semantics) *If $\vdash$ credits no more than the joint reading, every gap is a gap under it: if $\mathrm{Gap}(D)$ then no $\boldsymbol{b}$ has $\boldsymbol{b} \vdash D$. The converse fails: there is a corpus, such a reading and a demand that is a gap under the reading but covered under the joint one.*

So uncovered combinations reported under the joint reading are a subset of those any stricter model would report, while coverage claims enjoy no such protection. The same holds at the level of items: a benchmark gap is a gap for every item-level reading (`ItemCorpus.gapUnder_itemwise_of_gap`, `ItemCorpus.gapUnder_of_items_below`), while a benchmark can cover $\{a, b\}$ although none of its items does (`ItemCorpus.bench_covers_no_item_does`). Both directions are conditional on the assigned labels: a missed label makes a reported gap anti-conservative.

A.3 Resolution: counting at one level of detail

A labeller may name a card together with its parent. Counting over such a set counts one capability twice, so counting happens at a chosen level.

Definition 11 (Working cut and projection) *A* working cut *is a finite antichain $W \subseteq \mathcal{K}$ (no card of W lies below another). The* projection *of a demand onto W is $\pi_W(D) = \{w \in W : \exists c \in D,\ c \le w\}$, the cut cards D settles. A cell $X \subseteq W$ is* covered at W *if $X \subseteq \pi_W(d(\boldsymbol{b}))$ for some $\boldsymbol{b}$.*

Proposition 12 (Projection) (i) *$\pi_W(D)$ is an antichain, so $|\pi_W(D)|$ counts distinct capabilities.* (ii) *$D \sqsupseteq D'$ implies $\pi_W(D') \subseteq \pi_W(D)$.* (iii) *If $D \subseteq E$ and every card of E lies above some card of D, then $\pi_W(E) = \pi_W(D)$; in particular closing a demand upward does not change any projection.* (iv) *A card covered at W is covered: analysing at a cut invents no coverage.*

Rungs. A *stratification* is a level map $\ell : \mathcal{K} \to \mathbb{N}$ that strictly increases towards the specific end: $c \le c'$ and $c \ne c'$ imply $\ell(c') < \ell(c)$. The level is declared data, not a computed depth, because in a hierarchy where a card has two parents the two routes to the top may differ in length. Each *rung* $R_n = \{c : \ell(c) = n\}$ is then a working cut — the antichain condition is derived, not assumed — and no combination at a rung spans two levels. The stratification is *rooted* if every card with $\ell(c) > 0$ has a strictly broader card c^+ with $\ell(c^+) = \ell(c) - 1$.

Proposition 13 (Rooted rungs lose no demand) *If the stratification is rooted, every card has an ancestor at every shallower rung, and a demand containing a card at level $\ge n$ has a non-empty projection onto R_n. Without depth this fails: a demand whose cards all sit above rung n projects to $\emptyset$ on R_n.*

Changing the resolution. Cut W' is *finer* than W if every card of W' lies below a card of W; it *resolves* W if every card of W has a card of W' below it; and it is *unambiguous* if no card of W' lies below two distinct cards of W.

Theorem 14 (Coarsening only adds coverage; refining only reveals holes) *Let W' be finer than W, $w \in W'$, $v \in W$ and $w \le v$. If $\{w\}$ is covered at W' then $\{v\}$ is covered at W; equivalently, if v is a single-card gap at W, then so is every admissible $w \le v$ in W'. If moreover W' resolves W, is unambiguous, and every card of W' is admissible, the number of single-card gaps at W' is at least the number at W. The count has no invariant value, however: there is a corpus with one benchmark and two cuts, one finer than the other, with zero gaps at the coarse cut and two at the fine one.*

A bare count of empty cells is therefore a property of the chosen resolution, not of the corpus. What it keeps is a direction: taken at a coarse cut it is a lower bound on the count at any finer cut that resolves it unambiguously. The released rungs do not meet these hypotheses: 28 of the 180 coarse cards lie below more than one umbrella card and 55 of the 531 fine cards below more than one coarse card, so the rungs are not unambiguous, and 39 coarse cards have no fine card below them, so the fine rung does not resolve the coarse one. The count bound applies only within an unambiguous sub-forest, and we do not apply it to the released rungs.

Why a card is uncovered. An uncovered card c is *unresolved* if some broader card $c' \ge c$ is covered — a benchmark reaches the region but no label says where in it — and a *confirmed gap* if no card $c' \ge c$ is covered. Every card is exactly one of covered, unresolved, or a confirmed gap, and confirmed gaps are upward closed along $\le$. A confirmed gap is a hole in the ecosystem; an unresolved card may be a labelling defect. The distinction is decided at the umbrella rung alone

(`confirmedGap_iff_maximal`): in a rooted vocabulary whose umbrella cards are all covered there is no confirmed gap (`not_confirmedGap_of_rooted`). That is the case here, so all 203 uncovered cards of the 754 are unresolved. Confirmed-gap status is stable under refinement card by card (`confirmedGap_mono_le`), but the number of confirmed gaps is not grain-invariant (`confirmedGap_count_not_grain_invariant`).

A.4 What the labels leave undetermined

The labels we release are binary: a machine label is in or out at its threshold, and each human annotator answers yes or no. Undetermined entries arise nonetheless, in two ways. Where two annotators labelled the same benchmarks, the pairs they disagree on have no single answer; and a coarse label is silent about which finer card it stands for. The model records such an entry as $?$, so a row becomes a vector over $\{1, 0, ?\}$ — a device for reasoning about disagreement and coarseness, not a third output of any labeller.

Definition 15 (Partial corpus and its readings) *A* partial corpus *assigns each benchmark two demands $s(\boldsymbol{b}) \subseteq p(\boldsymbol{b})$: the cards asserted (*sure*) and the cards not denied (*possible*); an entry is $?$ iff the card is in $p(\boldsymbol{b}) \setminus s(\boldsymbol{b})$. Its* lower *reading is the corpus s (every silence a denial) and its* upper *reading the corpus p (every silence an assertion). A* completion *is a corpus d with $d(\boldsymbol{b}) \sqsupseteq s(\boldsymbol{b})$ and $p(\boldsymbol{b}) \sqsupseteq d(\boldsymbol{b})$ for every $\boldsymbol{b}$.*

The completion condition uses the demand order rather than containment on purpose. A coarse label says the required capability lies *somewhere under* a card, so a benchmark labelled `reasoning` whose true demand is `deductive-reasoning` must count as consistent with its label; $s(\boldsymbol{b}) \subseteq d(\boldsymbol{b})$ would reject it.

For a finite family $\mathcal{D}$ of demands and a corpus d, let $N_d(\mathcal{D}) = |\{D \in \mathcal{D} : \mathrm{Cov}_d(D)\}|$, the number of demands of the family d covers.

Theorem 16 (Coverage under partial labels is an interval) *For every completion d of a partial corpus and every finite family $\mathcal{D}$,*

$$N_s(\mathcal{D}) \;\le\; N_d(\mathcal{D}) \;\le\; N_p(\mathcal{D}),$$

in particular for the family $\binom{\Phi_0}{k}$ of k-subsets of a vocabulary Φ_0, at every order k. Both s and p are themselves completions, so neither bound can be tightened without more evidence; and D is covered under p but not under s exactly when two completions disagree about whether D is covered.

Only demands outside the band have an answer independent of how the silences resolve. The unresolved cards of the previous paragraph are the same phenomenon: recording a coarse label's silence about a finer card as $?$ turns every unresolved card into one in the band.

Two annotators form a partial corpus. Given two independent label sheets $L_1, L_2 : \mathcal{B} \to 2^{\mathcal{K}}$, set $s = L_1 \cap L_2$ and $p = L_1 \cup L_2$. The lower reading is then the *unanimous* merge, the upper reading the *union* merge, and the $?$ entries are exactly the (benchmark, card) pairs on which the annotators disagree. The choice of merge rule is therefore a choice of endpoint of the interval in Theorem 16, not a preprocessing step. The interval contains the true count only if the true demand lies between the two sheets; agreement between the annotators does not establish this, since both can miss a demand.

A.5 Combinations

Fix a working cut W. The *covered family at order k* is

$$\mathcal{C}_k = \{X \subseteq W : \; |X| = k, \; X \text{ covered at } W\} \;=\; \bigcup_{\boldsymbol{b} \in \mathcal{B}} \binom{\pi_W(d(\boldsymbol{b}))}{k},$$

the union over benchmarks of the k-subsets of each projection; write $\beta_{\boldsymbol{b}} = |\pi_W(d(\boldsymbol{b}))|$ for the breadth of $\boldsymbol{b}$ at W.

Proposition 17 (Arithmetic ceiling) *$|\mathcal{C}_k| \le \sum_{\boldsymbol{b}} \binom{\beta_{\boldsymbol{b}}}{k}$; if $\beta_{\boldsymbol{b}} < k$ for every $\boldsymbol{b}$ then $\mathcal{C}_k = \emptyset$; and the shadow of $\mathcal{C}_k$ lies in $\mathcal{C}_{k-1}$, so $\bigcup_k \mathcal{C}_k$ is a simplicial complex on W.*

Decay of coverage with k is thus partly forced by the breadth of the instruments, and a finding about decay has to beat this ceiling. Because the covered family is a simplicial complex, the Kruskal–Katona theorem constrains its counts at consecutive orders against each other with no null model at all. Write every $a \in \mathbb{N}$ in its r-cascade form $a = \binom{a_r}{r} + \binom{a_{r-1}}{r-1} + \cdots + \binom{a_j}{j}$ with $a_r > a_{r-1} > \cdots > a_j \ge j \ge 1$, and set $\partial_r(a) = \sum_i \binom{a_i}{i-1}$ and $\partial_r^{\uparrow}(a) = \sum_i \binom{a_i}{i+1}$.

Theorem 18 (Kruskal–Katona envelope on the covered family) *For every $k \ge 1$, $\partial_{k+1}\big(|\mathcal{C}_{k+1}|\big) \le |\mathcal{C}_k|$, equivalently*

$$|\mathcal{C}_{k+1}| \;\le\; \partial_k^{\uparrow}\big(|\mathcal{C}_k|\big),$$

the two forms being a Galois connection between ∂ and $\partial^{\uparrow}$. The cap is attained: whenever it does not exceed $\binom{N}{k+1}$ there is a family of exactly $\partial_k^{\uparrow}(a)$ sets of size $k+1$ in $[N]$ whose shadow has at most a elements.

We therefore report coverage at order $k+1$ as a fraction of the cap its order-k coverage permits (the *fill*); because the cap is attained, the denominator is a reachable configuration rather than a loose bound.

Filled and empty regions. At an evidence bar $q \ge 1$, a cell $X \subseteq W$ is *filled*, $\mathrm{Fill}_q(X)$, if at least q benchmarks cover it at W, and *empty* otherwise; $q = 1$ is coverage at W. Filled cells are closed under taking subsets and empty cells under taking supersets, so the filled region is exactly the subsets of its maximal filled cells and the empty region exactly the supersets of its minimal empty ones. A description of the two borders is therefore a lossless description of the whole space at every order at once, which is what the box summaries of the analysis compress.

A margin-preserving null. To test concentration beyond these constraints we compare against random corpora with the same margins. A *trade* between two benchmarks replaces their demands D_1, D_2 by D_1', D_2' with $|D_1'| = |D_1|$, $D_1' \cup D_2' = D_1 \cup D_2$ and $D_1' \cap D_2' = D_1 \cap D_2$ (the Curveball move). Every trade preserves the breadth of both benchmarks and the number of benchmarks demanding each card, so a chain of trades holds both margins of the benchmark–card incidence matrix fixed.

A.6 Conditioning on dials and probe regimes

A covered combination says nothing about the conditions it was covered under: a pair of capabilities demanded only by text-only benchmarks, or never under any probe regime, is covered in the tables and narrow in fact. The dials and the probe regimes record those conditions, and conditioning on them needs no new machinery. Let $\mu : \mathcal{B} \to 2^V$ assign each benchmark a finite set of values — the values of a multi-valued dial, or the probe regimes the benchmark applies; $\mu(\boldsymbol{b}) = \emptyset$ means $\boldsymbol{b}$ is *silent*. For a value v, the sub-corpus $\mathcal{B}_v = \{\boldsymbol{b} : v \in \mu(\boldsymbol{b})\}$ keeps each benchmark's demand and only chooses which benchmarks count; write $\mathrm{Cov}_{|v}$ for its coverage.

Theorem 19 (Conditioning is restriction) (i) *$\mathrm{Cov}_{|v}(D)$ iff some $\boldsymbol{b}$ with $v \in \mu(\boldsymbol{b})$ has $d(\boldsymbol{b}) \sqsupseteq D$, and the same holds at a working cut.* (ii) *Conditioning never adds coverage: $\mathrm{Cov}_{|v}(D)$ implies $\mathrm{Cov}(D)$, so every gap is a gap under every value.* (iii) *If no benchmark is silent, $\mathrm{Cov}(D)$ iff $\mathrm{Cov}_{|v}(D)$ for some v. With silence only the backward direction holds: there is a corpus and a demand that is covered and covered under no value.* (iv) *D is covered* only under *v — covered, and every benchmark that covers it has $v \in \mu(\boldsymbol{b})$ — iff $\mathcal{B}_v$ covers D and $\mathcal{B} \setminus \mathcal{B}_v$ does not.*

Part (iii) is why a pair supported by a silent benchmark is reported as *undetermined* rather than as narrow or broad, and part (iv) is the narrowness the analysis counts. Since conditioning selects benchmarks and never alters a demand, its null is a random sub-corpus of the same size, not a reshuffled demand.

Table 4: Objects of the model and what carries them in the vocabulary and the labels (16,217 benchmarks).

model	carried by	here
Φ (capabilities)	cards of kind facet	754 cards
T (probe regimes)	cards of kind probe regime	17 cards, flat
M (dials)	cards of kind dial, with a value set	14 axes
$\leq$	narrower / broader relations	within Φ only
ℓ, rungs R_n	declared level: umbrella, coarse, fine, atom	16 / 180 / 531 / 27
$d(\boldsymbol{b}) \cap \Phi$	vote $\hat{y}(\boldsymbol{b}, c)$; $c \in d(\boldsymbol{b})$ iff $\hat{y}(\boldsymbol{b}, c) \geq \tau$	$\tau = 0.47$
μ for each dial	parsed card fields, no threshold	14 dials
μ for the probe regimes	vote per card, its own cut	$\tau = 0.1125$
(s, p) of a partial corpus	two human sheets, $L_1 \cap L_2$ and $L_1 \cup L_2$	100 gold benchmarks

A.7 How the data instantiates the model

Table 4 gives the correspondence. The vocabulary is a list of cards; each card's declared kind fixes its kind in the model, and its narrower and broader relations generate $\leq$ by reflexive–transitive closure, never across kinds; only capability cards carry such relations, so T and M are discrete. Each capability card declares its rung, read as $\ell = 0, 1, 2, 3$ from umbrella to atom, which is the stratification of the model; the four rungs are the working cuts at which we count. The hypotheses the theorems use are checked against the vocabulary on every run rather than assumed: the order is antisymmetric, each rung is an antichain, ℓ is strictly monotone along all comparable pairs, and the stratification is rooted (no orphan card outside the umbrella rung).

The capability labels record, for each benchmark, a voted score $\hat{y}(\boldsymbol{b}, c) \in [0, 1]$ for every capability card; the demand is the set of cards at or above the operating threshold τ. Upward closure of that set is not required, since by Proposition 12(iii) it does not change any projection. Every coverage figure is computed on these demands. The dials and the probe regimes are not added to demands; they enter through Theorem 19, each as a map μ. Adding dial cards to demands would be equivalent (`Corpus.withDials`, `coveredClosed_withDials_iff`); there `Card.dial v` is an analytical encoding of a dial value, distinct from the catalogue card that declares the dial. Dial values are benchmark-level: a benchmark has value v if any of its items does. Under this aggregation a gap conditioned on a dial is sound (`ItemCorpus.gap_given_toItems_of_gap`), but narrowness is not: a pair can be narrow under vision when only text items demand it (`ItemCorpus.narrowUnder_any_not_items`), and with multi-valued dials it can be narrow under two values at once (`narrowUnder_two_values`); a sound certificate needs all-item or single-valued benchmarks (`ItemCorpus.narrowUnder_toItems_of_all`). A dial value is parsed from the benchmark's card. The parse fills some unstated dials with defaults (probe construction naturalistic on 15,163 records, artefact type on 7,963, language English on 3,713, subject domain on 2,097, input modality text on 2,048), which the analysis treats as known values; the Lean observation layer keeps known, unknown, inapplicable, unresolved and conflicting states apart and projects only known ones to values (`Observations.toMultiDial`). A benchmark's probe regimes are the tags its row carries; a benchmark with none is recorded as having no probe regime tagged. With probe-regime recall about 0.52, that is missing evidence, not a statement that the benchmark applies none. The probe-regime labels are weak (precision 0.30 on their gold set) and over-assign: a benchmark carrying many tags is most likely wrong rather than a benchmark that applies them all. Any result conditioned on probe regimes should therefore be reported twice, as released and with each benchmark carrying five or more tags treated as silent, which by Theorem 19(iii) makes the pairs it supports undetermined; this paper reports no such result. Where two human annotators labelled the same benchmarks, their sheets are combined into a partial corpus as above, and every figure computed on that subset is reported as the interval of Theorem 16. The machine labels carry no ? entry: for them $s = p$, the interval collapses to a point, and the corpus-wide figures are point estimates, not bounds on the true demands.

B THE FORMAL MODEL IN DETAIL

This appendix completes Appendix A: it gives the definitions the main text only names, a proof or proof sketch for each stated result, and the Lean 4 declaration that carries it. The development lives in `formal/` (module `BenchmarkFacet`), depends on Mathlib and nothing else, and builds with `lake build`. Where a proof is more than a few lines we give only its idea; the Lean source is the full proof. All declarations cited below reduce to Lean's three standard axioms (`propext`, `Classical.choice`, `Quot.sound`); no `sorry` appears, and a build step (`BenchmarkFacet.FinalCheck`) prints the axioms of each cited declaration and fails if any changes.

Throughout, $\mathcal{K}$ is a type of cards with a preorder $\leq$, demands are finite subsets of $\mathcal{K}$, a corpus is a pair $(\mathcal{B}, d)$ with $d : \mathcal{B} \to 2^{\mathcal{K}}$, and $\boldsymbol{b}$ ranges over benchmarks.

B.1 TYPING

Definition 20 (Setting, dial, probe regime, capability) *A* setting *is a pair* $(\alpha : \mathcal{B} \to \mathrm{Ans},\ \sigma : \mathcal{A} \to \mathcal{B} \to \mathrm{Score})$. *A* dial *with values* V *is a function* $\mu : \mathcal{B} \to V$. *A* probe regime *is a map* $\rho : \mathcal{B} \to \mathcal{B}$ *with* $\alpha(\rho(\boldsymbol{b})) = \alpha(\boldsymbol{b})$ *for all* $\boldsymbol{b}$. *A* capability *is a predicate* κ *on* $\mathcal{A}$. *A probe regime is* invariant *for the setting if* $\sigma(\boldsymbol{a}, \rho(\boldsymbol{b})) = \sigma(\boldsymbol{a}, \boldsymbol{b})$ *for all* $\boldsymbol{a}, \boldsymbol{b}$ *(the INV relation of metamorphic testing (Chen et al., 1998; Ribeiro et al., 2020)).*

Probe regimes form a monoid under composition with the identity as unit; composition preserves truth because each factor does. Nothing makes them a group: padding with distractors composes with a paraphrase, but padding has no inverse in general (`Perturbation.exists_not_invertible`). The monoid is a property of the abstraction, not closure of any catalogue of regimes. A probe regime need not be invariant: it may change the score, and it may destroy a test — a benchmark that separates two models may stop separating them once the regime is applied. The type is the answer-preserving case only; outcome relations other than identity and the task-contract conditions of a real probe are outside it (Appendix B.8).

Proof of Proposition 6. Take $\mathcal{A} = \{\top, \bot\}$, one benchmark $\boldsymbol{b}$, a constant answer key, and $\kappa = \{\top\}$. In the first setting $\sigma(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}$, so $\top$ and $\bot$ are separated and $\boldsymbol{b}$ tests κ. In the second the score is constantly $\top$, so no pair is separated. The two settings share benchmarks and answer key, so any function of the artefact takes the same value on both. □

Proof of Proposition 7. In the first setting above take $\kappa' = \{\bot\}$. The pair $(\top, \bot)$ witnesses $\Theta(\boldsymbol{b}, \kappa)$ and the pair $(\bot, \top)$ witnesses $\Theta(\boldsymbol{b}, \kappa')$, while κ and κ' disagree on both models. With a single benchmark, both are tested by exactly the same benchmarks. □

Definition 21 (Card space) $\mathcal{K} = \Phi \sqcup T \sqcup M$ *with injections* $\iota_\Phi, \iota_T, \iota_M$ *and the order* $\iota_\Phi(x) \leq \iota_\Phi(y)$ *iff* $x \leq_\Phi y$; $\iota_T(x) \leq \iota_T(y)$ *iff* $x \leq_T y$; $\iota_M(x) \leq \iota_M(y)$ *iff* $x = y$; *and no relation between cards of different kinds.*

This is a preorder; comparable cards have the same kind; dial cards are comparable only to themselves; and a capability card lies below another exactly when the capabilities do. The independence of the axes is thus a design decision recorded as the shape of the order, not derived. The order on T is any preorder; the released taxonomy takes the discrete one, on which the demand order is reverse inclusion exactly as for dials.

B.2 DEMANDS, COVERAGE AND GAPS

The demand order of Definition 8 is reflexive and transitive: if every card of D'' has a card of D' below it and every card of D' one of D below it, composing the witnesses gives one of D below each card of D''. If $\leq$ is discrete then a witness $c \leq c'$ forces $c = c' \in D$, so $D \sqsupseteq D'$ iff $D' \subseteq D$. Every demand demands $\emptyset$, and only $\emptyset$ is demanded by $\emptyset$.

Definition 22 (Native coverage, completeness, minimal gap) D *is* covered natively *if* $d(\boldsymbol{b}) = D$ *for some* $\boldsymbol{b}$. *The corpus is* complete *for a vocabulary* Φ_0 *if every* $c \in \Phi_0$ *has a* $\boldsymbol{b}$ *with* $d(\boldsymbol{b}) = \{c\}$

— the Q-matrix completeness condition of Liu et al. (2012). A gap D is minimal *if every admissible D' with $D \sqsupseteq D'$ and $D' \not\sqsupseteq D$ is covered.*

Native coverage implies coverage and the converse fails (one benchmark demanding $\{0,1\}$ covers $\{0\}$ but not natively). A complete corpus covers every card of its vocabulary natively. A minimal gap is an inclusion-minimal uncovered interaction: the uncovered combination immediately above the covered region. It is not a "forbidden tuple" in the sense of combinatorial interaction testing, which means infeasible and is what adm rules out.

Coverage is downward closed, gaps rise. If $d(\boldsymbol{b}) \sqsupseteq D$ and $D \sqsupseteq D'$ then $d(\boldsymbol{b}) \sqsupseteq D'$ by transitivity. A gap D' with $D \sqsupseteq D'$ and $\mathrm{adm}(D)$: were D covered, D' would be. □

Definition 23 (Combination semantics) *A* semantics *is a relation $\boldsymbol{b} \vdash D$ between benchmarks and demands; D is covered under it if $\boldsymbol{b} \vdash D$ for some $\boldsymbol{b}$, and a gap under it if* $\mathrm{adm}(D)$ *and it is not. It* credits no more *than the joint reading if $\boldsymbol{b} \vdash D \Rightarrow d(\boldsymbol{b}) \sqsupseteq D$.*

The joint reading $\boldsymbol{b} \vdash D :\Leftrightarrow d(\boldsymbol{b}) \sqsupseteq D$ is itself such a semantics, and coverage under it is Cov.

Proof of Theorem 10. If $\boldsymbol{b} \vdash D$ then $d(\boldsymbol{b}) \sqsupseteq D$, so $\mathrm{Cov}(D)$, contradicting the gap. For the converse, take one benchmark with $d(\boldsymbol{b}) = \{0,1\}$, the semantics $\boldsymbol{b} \vdash D :\Leftrightarrow D = \{0,1\}$, trivial admissibility, and $D = \{0\}$: the semantics credits no more than the joint reading, $\{0\}$ is not credited under it, and $\{0,1\} \sqsupseteq \{0\}$ covers it jointly. □

B.3 Resolution

Proof of Proposition 12. (i) $\pi_W(D) \subseteq W$ and W is an antichain. (ii) If $w \in \pi_W(D')$ then $D' \sqsupseteq \{w\}$, and $D \sqsupseteq D'$ gives $D \sqsupseteq \{w\}$. (iii) $\pi_W(D) \subseteq \pi_W(E)$ by (ii). Conversely, if $c \in E$, $c \le w$, pick $e \in D$ with $e \le c$; then $e \le w$. (iv) If $\{w\} \subseteq \pi_W(d(\boldsymbol{b}))$ then some $c \in d(\boldsymbol{b})$ has $c \le w$, i.e. $d(\boldsymbol{b}) \sqsupseteq \{w\}$. □

At a fixed cut, coverage is downward closed under inclusion, and gaps are upward closed.

Definition 24 (Stratification, rootedness, no rung skip) *A* stratification *of $\mathcal{K}$ is $\ell : \mathcal{K} \to \mathbb{N}$ with $c \le c'$, $c \ne c' \Rightarrow \ell(c') < \ell(c)$. For a finite vocabulary $K_0 \subseteq \mathcal{K}$ the rung $R_n = \{c \in K_0 : \ell(c) = n\}$. The stratification is* rooted *on K_0 if every $c \in K_0$ with $\ell(c) \ne 0$ has $c^+ \in K_0$ with $c < c^+$ and $\ell(c^+) + 1 = \ell(c)$. It has* no rung skip *if every $c < c^+$ in K_0 either has $\ell(c^+) + 1 = \ell(c)$ or factors through some $c < c'' < c^+$ in K_0.*

A stratification is weaker than a graded order in Mathlib's sense, which additionally requires every cover to step the grade by one; no rung skip is exactly that extra condition, and it implies that every covering pair steps the level by one. Neither notion implies rootedness, which is the hypothesis the results below use: a graded order constrains a cover *if one exists* but never asserts that a non-top card has one.

Rungs are cuts. If $c, c' \in R_n$ and $c \le c'$ with $c \ne c'$, then $\ell(c') < \ell(c) = \ell(c')$. Consequently a subset of a rung lies at one level, and the rungs partition K_0. □

Proof of Proposition 13. By induction on $\ell(c) - n$: rootedness gives a parent one level up, and composing $\le$ along the chain reaches level n. If $c \in D$ has $\ell(c) \ge n$, its ancestor c^+ at level n lies in $\pi_{R_n}(D)$. Conversely if every $c \in D$ has $\ell(c) < n$ and $c \le w$ with $\ell(w) = n$, then $c = w$ is impossible and $c \ne w$ gives $\ell(w) < \ell(c) < n$. □

Proof of Theorem 14. Direction. If $\{w\} \subseteq \pi_{W'}(d(\boldsymbol{b}))$ then some $c \in d(\boldsymbol{b})$ has $c \le w \le v$, so $v \in \pi_W(d(\boldsymbol{b}))$; the gap statement is the contrapositive. *Count.* Let G and G' be the single-card gaps at W and W'. Since W' resolves W, choose for each $v \in W$ some $f(v) \in W'$ with $f(v) \le v$. By unambiguity f is injective: $f(v_1) = f(v_2)$ lies below both v_1 and v_2. By the direction statement and admissibility, f maps G into G'. Hence $|G| \le |G'|$. *No invariant value.* Take four cards, one broad card c_0 and three pairwise incomparable specialisations $f_0, f_1, f_2 \le c_0$, and one benchmark with $d(\boldsymbol{b}) = \{f_0\}$. At the cut $\{c_0\}$ nothing is uncovered; at the finer cut $\{f_0, f_1, f_2\}$ both f_1 and f_2

are. Neither is a confirmed gap, since $c_0 \geq f_i$ is covered; the confirmed-gap count is zero at both cuts. □

Definition 25 (Gap status) *An uncovered card c is* unresolved *if some $c' \geq c$ has $\mathrm{Cov}(\{c'\})$, and a* confirmed gap *if no $c' \geq c$ does.*

Trichotomy and monotonicity. If $\{c\}$ is uncovered, either some $c' \geq c$ is covered or none is. If c is a confirmed gap and $c \leq c'$, every $c'' \geq c'$ is $\geq c$, so c' is a confirmed gap. The converse direction fails in general — a card may sit under two unrelated broader cards — and what survives downward is only that a card below a confirmed gap is uncovered. □

B.4 Partial observation

A partial corpus is $(\mathcal{B}, s, p)$ with $s(\boldsymbol{b}) \subseteq p(\boldsymbol{b})$; its entry at $(\boldsymbol{b}, c)$ is 1 if $c \in s(\boldsymbol{b})$, ? if $c \in p(\boldsymbol{b}) \setminus s(\boldsymbol{b})$, and 0 otherwise, so the pair and the $\{0, 1, ?\}$ incidence matrix determine each other. We distinguish two notions of consistent true demand:

- a *containment completion*: $s(\boldsymbol{b}) \subseteq d(\boldsymbol{b}) \subseteq p(\boldsymbol{b})$ for all $\boldsymbol{b}$;
- a *refinement completion* (the notion used in the main text): $d(\boldsymbol{b}) \sqsupseteq s(\boldsymbol{b})$ and $p(\boldsymbol{b}) \sqsupseteq d(\boldsymbol{b})$ for all $\boldsymbol{b}$.

Every containment completion is a refinement completion, and the inclusion is strict: with cards $3 \leq 5$, a benchmark with $s = \{5\}$, $p = \{3, 5\}$ and true demand $\{3\}$ is a refinement completion but not a containment one. This case, where the possible reading names the finer card, is why the bounds are stated for the weaker hypothesis. It does not cover a label coarser than the true demand: a benchmark labelled `reasoning` whose true demand is a finer reasoning card is not a refinement completion, and the upper bound can fail for it (`refinement_below_label_escapes_upper`); the bound holds again when the undenied cards form a down-closed set (`coveredClosed_upper_of_downClosed`). Likewise, the intersection of two raters is not a sure reading when they label at different grains (`lower_ofTwoAnnotators_grain_mismatch`). The upper side is modelled only through $\sqsupseteq$; the dual (Hoare) constraint "every demanded card lies below something undenied" is not modelled. The three-valued row is Rubin's separation of data from missingness indicator (Rubin, 1976); the model bounds rather than imputes, so it needs no assumption about how the silences arose. The bounds hold only if the true assignment lies within the envelope: they do not protect against both raters missing a target, both asserting a wrong one, or benchmarks missing from the corpus.

A demand is *certainly covered* if the lower reading covers it, *possibly covered* if the upper one does, *undetermined* if possibly but not certainly covered, and a *certain gap* if not possibly covered. Certain gaps are upward closed along $\sqsupseteq$ by the same argument as gaps.

Proof of Theorem 16. Soundness. If $s(\boldsymbol{b}) \sqsupseteq D$ then $d(\boldsymbol{b}) \sqsupseteq s(\boldsymbol{b}) \sqsupseteq D$; if $d(\boldsymbol{b}) \sqsupseteq D$ then $p(\boldsymbol{b}) \sqsupseteq d(\boldsymbol{b}) \sqsupseteq D$. So every demand covered under s is covered under d, and every demand covered under d under p. Counting a filtered finite family is monotone in the filter, which gives the two inequalities for any $\mathcal{D}$, and for $\mathcal{D}$ the k-subsets of Φ_0 in particular. *Attainment.* s and p satisfy both containments, so both are completions. *The band.* If D is undetermined, p and s are two completions that disagree. Conversely, if completion d covers D and d' does not, soundness gives that p covers D and that s does not. (The Lean statement of the band is for containment completions; the refinement version is the same argument using the two soundness lemmas, which are stated at the refinement hypothesis.) □

Two annotators. With $s = L_1 \cap L_2$ and $p = L_1 \cup L_2$, $s \subseteq p$, and by definition the lower and upper corpora are the unanimous and union merges. An entry is ? iff $c \in (L_1 \cup L_2) \setminus (L_1 \cap L_2)$, i.e. exactly one annotator asserted c. □

Unresolved cards are undetermined. Given a corpus d and a card c, let the partial corpus have $s = d$ and $p(\boldsymbol{b}) = d(\boldsymbol{b}) \cup \{c\}$, recording the labels' silence about c as ?. If c is unresolved, $\{c\}$ is not covered under $s = d$ but is covered under p (by c itself), so it is undetermined. □

B.5 COMBINATIONS

Proof of Proposition 17. $X \in \mathcal{C}_k$ iff $|X| = k$ and $X \subseteq \pi_W(d(\boldsymbol{b})) \subseteq W$ for some $\boldsymbol{b}$, which is the displayed union; the union bound gives the ceiling, and a k-set cannot fit in a projection of size $\beta_{\boldsymbol{b}} < k$. Removing one element of $X \subseteq \pi_W(d(\boldsymbol{b}))$ leaves a $(k-1)$-subset of the same projection, so the shadow of $\mathcal{C}_k$ lies in $\mathcal{C}_{k-1}$, and iterating gives $\partial^i \mathcal{C}_k \subseteq \mathcal{C}_{k-i}$. □

Cascades. For $r \geq 1$ and $a \in \mathbb{N}$ let a_r be the largest x with $\binom{x}{r} \leq a$; repeating on $a - \binom{a_r}{r}$ at $r-1$ down to $r = 0$ produces the r-cascade cascade (Macaulay) representation, i.e. the combinatorial number system. ∂_r and $\partial_r^{\uparrow}$ of the main text lower, respectively raise, the bottom index of every term. Both are monotone; $\partial_r^{\uparrow}(a) \leq \binom{N}{r+1}$ whenever $a \leq \binom{N}{r}$; and they form a Galois connection, $\partial_{r+1}(b) \leq a \iff b \leq \partial_r^{\uparrow}(a)$ for $r \geq 1$. They are strictly smaller than the multicomplex shift the same representation carries for Macaulay's theorem, which raises both indices of each term and is not a valid cap for set families (three edges admit one triangle; that shift permits four).

Proof sketch of Theorem 18. The Kruskal–Katona theorem (Kruskal, 1963; Katona, 1968) in cascade form states that a family $\mathcal{F}$ of r-sets has $|\partial\mathcal{F}| \geq \partial_r(|\mathcal{F}|)$. Mathlib proves it only for subsets of $\operatorname{Fin} n$, via colex initial segments. We (a) define the colex rank of an r-set of $\mathbb{N}$ and show that the first m r-sets in colex order have shadow of size exactly $\partial_r(m)$; (b) combine this with Mathlib's statement that an initial colex segment minimises the shadow to get the cascade form on $\operatorname{Fin} n$; and (c) transport it to an arbitrary finite ground set W by an embedding $\operatorname{Fin}|W| \hookrightarrow W$, pulling the family back and pushing the shadow forward (as a subset, which is all a lower bound needs). Applied to $\mathcal{F} = \mathcal{C}_{k+1}$, whose shadow lies in $\mathcal{C}_k$ by Proposition 17, this gives $\partial_{k+1}(|\mathcal{C}_{k+1}|) \leq |\mathcal{C}_k|$, and the Galois connection turns it into the cap. Attainment: the first $\partial_k^{\uparrow}(a)$ sets of size $k+1$ of $[N]$ in colex order have exactly that many elements, and by (a) and the adjunction their shadow has at most a. The weaker Lovász form (Lovász, 1979) — $|\mathcal{F}| \geq \binom{x}{r}$ implies $|\partial^i\mathcal{F}| \geq \binom{x}{r-i}$ — is proved the same way and gives the corollary that $|\mathcal{C}_{k-i}| < \binom{N}{k-i}$ forces $|\mathcal{C}_k| < \binom{N}{k}$. □

The adjunction covers every order from $k+1 = 2$ upward. There is no cap on order one from order zero: the only 0-set is $\emptyset$, and a family of singletons of any size has shadow $\{\emptyset\}$.

Trades. A trade $(D_1, D_2) \mapsto (D_1', D_2')$ requires $|D_1'| = |D_1|$, $D_1' \cup D_2' = D_1 \cup D_2$ and $D_1' \cap D_2' = D_1 \cap D_2$. Then $|D_2'| = |D_2|$ follows from $|X \cup Y| + |X \cap Y| = |X| + |Y|$, and a card lies in both, one or none of D_1', D_2' exactly as it did in D_1, D_2, so its column sum is unchanged; benchmarks other than the two traded are untouched. The randomisation is the Curveball move (Strona et al., 2014); we prove the invariant, not that the chain mixes, which is checked empirically.

Filled and empty regions. If $X' \subseteq X$, every benchmark covering X at W covers X', so $\operatorname{Fill}_q(X)$ implies $\operatorname{Fill}_q(X')$ and, contrapositively, emptiness rises along $\subseteq$; Fill_1 is coverage at W. Every filled cell lies under a maximal filled one and every empty cell over a minimal empty one (both families are finite, being subsets of 2^W), which is the border description. For the box summaries, a *box* is a pattern of k slots, each a card (any card at or below it matches) or a wildcard; if every cell a box matches is empty, every larger cell containing a match is empty too, which is what lets a box found at order k speak for all higher orders.

B.6 CONDITIONING

A multi-valued axis is $\mu : \mathcal{B} \to 2^V$ with finite values; the single-valued dial of the typing is the case of singletons. For a predicate P on benchmarks the *restriction* keeps the benchmarks satisfying P and their demands unchanged; $\mathcal{B}_v$ is the restriction to $v \in \mu(\boldsymbol{b})$.

Proof of Theorem 19. (i) Unfolding: a benchmark of the restriction is a $\boldsymbol{b}$ with $P(\boldsymbol{b})$, and its demand is $d(\boldsymbol{b})$; at a cut the same argument applies to projections. (ii) A witness in the restriction is a witness in $\mathcal{B}$; a gap has no witness in $\mathcal{B}$, hence none in any restriction. Restriction is also monotone in the predicate. (iii) If $d(\boldsymbol{b}) \sqsupseteq D$ and $\mu(\boldsymbol{b})$ is non-empty, pick $v \in \mu(\boldsymbol{b})$; then $\boldsymbol{b} \in \mathcal{B}_v$ witnesses $\operatorname{Cov}_{|v}(D)$. The converse is (ii). For the failure, take one benchmark with $d(\boldsymbol{b}) = \{0\}$ and $\mu(\boldsymbol{b}) = \emptyset$: $\{0\}$ is covered, and no $\mathcal{B}_v$ contains $\boldsymbol{b}$. (iv) If D is covered only under v, its witness lies in $\mathcal{B}_v$, and a witness

outside $\mathcal{B}_v$ would violate "only". Conversely a witness in $\mathcal{B}_v$ gives coverage, and any covering $\boldsymbol{b}$ with $v \notin \mu(\boldsymbol{b})$ would be a witness in the complement. □

The released probe-regime axis records a benchmark with no tag as having no probe regime tagged, so it is never silent as released, although the absence is missing evidence rather than a negative assignment; the defensive variant of the analysis makes the over-tagged benchmarks silent on purpose, and part (iii) is then what turns the pairs they support into undetermined ones.

B.7 Declarations

Table 5 maps each numbered statement to the Lean declarations that carry it; all names are in the namespace `BenchmarkFacet`. The `Examples` modules instantiate most public declarations on a small worked case, which rules out, for those, definitions that hold vacuously and theorems whose hypotheses nothing satisfies.

B.8 What the model does not contain

- **No model of how capabilities combine.** Whether co-assigned capabilities are exercised conjunctively or compensatorily needs item-response data the corpus does not have. Theorem 10 shows that fitting such a model cannot remove a gap; coverage claims are not protected in the same way.
- **No interaction between the axes.** The card order makes Φ, T and M independent by construction; whether a dial value rules out a capability is not modelled. Conditioning (Theorem 19) reads coverage within a value; it does not assert that the capability depends on it.
- **Probe regimes only as answer-preserving maps.** A probe regime in the full sense varies how items are presented relative to a reference, within a declared task contract, with a justified outcome relation on a stated domain. The type captures the invariance case, $\alpha \circ \rho = \alpha$ on all of $\mathcal{B}$. Directional tests and substitutions that map the answer, domain restrictions, the preservation of task mechanics and objective, and adaptive procedures are not encoded; nor is the link from a probe-regime *card* to a map ρ — as for capabilities, the labels record what a benchmark is documented to do.
- **No graded demands and no response model.** Demands are sets, so a label is in or out. Two extensions are natural; the first remains a conjecture, the second is now proved. (a) *Probabilistic labels.* If each pair $(\boldsymbol{b}, c)$ is targeted with probability $\pi(\boldsymbol{b}, c)$, every realisation should be a containment completion of the partial corpus with $s(\boldsymbol{b}) = \{c : \pi(\boldsymbol{b}, c) = 1\}$ and $p(\boldsymbol{b}) = \{c : \pi(\boldsymbol{b}, c) > 0\}$, so that by Theorem 16 each realisation's count, and hence the expected count under any distribution over realisations, lies in $[N_s(\mathcal{D}), N_p(\mathcal{D})]$. (b) *A confirmatory response model.* If each item i of a benchmark carries an attribute vector $q(i)$ with $d(\boldsymbol{b}) \sqsupseteq q(i)$, the conjunctive reading $\boldsymbol{b} \vdash D :\Leftrightarrow \exists i : q(i) \sqsupseteq D$ should credit no more than the joint reading, so that by Theorem 10 every gap survives it. (a) looks like a short corollary of `countCovered_completion_bounds` but is not in the Lean development; (b) is proved (`ItemCorpus.creditsNoMore_of_items_below`, `ItemCorpus.gapUnder_of_items_below`). Whether a fitted model identifies the tested capability (Proposition 7), and how well it fits, are statistical questions outside the model.
- **No causal reading of Θ.** The separation criterion is the observational counterpart of a causal validity criterion, which is what a corpus of scores can support.
- **No admissibility grammar.** adm is a parameter; the vocabulary states no cross-axis grammar from which it could be derived.
- **No parse of the released files.** The model starts from demands. That the label file parses to those demands, and that the order, rungs and rootedness hold on the released taxonomy, are checked by the analysis scripts at run time rather than proved; the Python implementation of the cascade operators is tested against brute force separately, and every cascade value the paper reports is additionally evaluated in the Lean kernel (`Examples.CascadeV13`).

C EXTENDED RELATED WORK

Benchmark assessment and evaluation science. HELM (Liang et al., 2023) and BIG-bench (Srivastava et al., 2023) established that evaluation should run across many scenarios under standard conditions. BetterBench (Reuel et al., 2024) audits benchmarks against a quality rubric, validity-centred frameworks (Salaudeen et al., 2025; Kearns, 2026) ask which construct a score is evidence for, and statistical work places benchmarks on a common difficulty scale (Ho et al., 2025). Weidinger et al. (2025) draw these threads into a call for an evaluation science for generative AI. All of this work improves what is concluded from benchmarks that exist. Our question is prior to it: what a coverage table over those benchmarks establishes, and how much of it follows from undeclared choices.

Cataloguing the ecosystem. The closest prior instrument is the Intelligence Task Ontology (Blagec et al., 2022), used to analyse reported metrics (Blagec et al., 2020) and benchmark saturation over time (Ott et al., 2022). Later work maps evaluation resources across the arXiv literature (Wang, 2026), induces capability taxonomies automatically (Wan et al., 2024; Wang et al., 2026), or locates concept-level gaps from model internals (Bohacek et al., 2026). Our emphasis is a reporting framework for the sensitivity and evidential limits of coverage computed from such maps. The closest recent study (Fang et al., 2026) maps research papers, not benchmarks, onto 14 cognitive domains, analyses one reading of its labels only, and does not report its labelling pipeline against its experts. At that grain, nearly every pair of capabilities is already co-assigned in our corpus, so combination gaps appear only at finer resolution (Appendix C).

Q-matrices and item response theory. The object we build is a benchmark-level analogue of a Q-matrix (Tatsuoka, 1983): an incidence relation between benchmark records and their documented demands. Item-level Q-matrices are studied in cognitive diagnosis with ordered attributes (Leighton et al., 2004) and with estimated entries (Liu et al., 2012; de la Torre, 2009). Evidence-centred design (Mislevy et al., 2003) asks that the chain from response to attribute be written down; our four choices are the links of that chain that benchmarking leaves implicit. Unlike cognitive diagnosis, we have no response table against which to check the matrix. Item response theory, now used in NLP (Lalor et al., 2024), would supply one, but a response can reflect a shortcut rather than the intended capability (Chollet, 2019). Our labels therefore record what a benchmark is built to demand, not what it was observed to test.

Joint exercise and LLM labellers. Skill-Mix (Yu et al., 2024) builds items that must exercise a random k-subset of skills (Arora & Goyal, 2023). It is the constructive complement of our result: joint exercise can be established by construction but not inferred from co-occurring labels. STEPS (Wei et al., 2026) learns a skill hierarchy from co-occurrence in training data to target rare combinations for synthesis. Our ladder is instead fixed and declared independently of the corpus, with every card defined and levelled, which is what lets absent capabilities appear as gaps and lets coverage be compared across resolutions, label readings and corpora. Metamorphic testing (Chen et al., 1998; Segura et al., 2016) and behavioural test suites (Ribeiro et al., 2020) show the same directionality. LLM judges carry systematic biases (Zheng et al., 2023; Li et al., 2026), which is why we treat the labeller as a measurement instrument with its own error band. Appendix C extends this discussion.

Benchmark assessment. Beyond the work in Section 1.1, surveys organise published benchmarks into capability taxonomies (Ni et al., 2025) or track how a score can be inflated (Chen et al., 2025), and statistical work shows that single-run scores understate what a model population can do (Fowler et al., 2026). Weidinger et al. (2025) ask for metrics tied to deployed performance, refined over time and maintained by standing institutions. A reporting standard for coverage claims is one of the concrete things such an evaluation science would have to contain.

Scope relative to model-internal gap analysis. Bohacek et al. (2026) locate concept-level gaps in models and in the benchmarks that test them from model-internal representations. We label what a benchmark demands from its documentation, which is available for the whole ecosystem, rather than what a particular model represents, which requires running that model.

Relation to Fang et al. (2026). Their taxonomy is built top-down from cognitive science: 14 domains in three layers (primitive, constructed, integrative) and 91 subskills, each subskill fixed under one domain. It is applied to 15,934 papers from four venues (2023–2025) by three LLM annotators, with a fourth model arbitrating the 27 % of screened papers without a two-of-three consensus. We share the aim of moving from tasks to capabilities and differ in what the result can support.

- *What is measured.* A paper that studies a capability is evidence of research attention; a benchmark that demands it is evidence of what can be evaluated. Their own limitations restrict their patterns to research attention in the chosen venues.
- *Validation of the labeller.* The single reliability figure is agreement between two experts on domain presence ($\kappa = 0.70$, 50 papers). No agreement between the pipeline or the arbiter and the experts is reported, and none for the subskills or for the strong/weak evidence split, although every analysis uses strong-evidence labels only. We report the labeller's precision, recall and F_1 against the gold set, leave one benchmark out, next to the agreement between the two human raters (Table 9).
- *Definitions.* Their definitions have a median of about 17 words, carry no examples, and only two or three of the 105 state a boundary against a neighbouring category; some describe human biology rather than model behaviour (sensory transduction as "converting physical sensory inputs into neural activity"). Each of our cards has positive and negative anchors and at least one contrastive example that names the nearest card and the rationale for the boundary (1,323 in total).
- *Structure.* Their layers encode hypothesised functional support, assigned by judgment; the lateral and vertical relations are drawn in a figure, and the paper states that it does not validate them. Our ladder is a subsumption order given as data: 85 cards have more than one parent, 161 sit under more than one umbrella, and 874 overlaps within a rung are declared. The obligations the proofs need are checked by machine.
- *Types.* Theirs is a capability-only vocabulary. Ours separates evaluative facets from probe regimes and setup dials, so coverage can be conditioned on how a benchmark is run (Theorem 19).
- *Reach of the vocabulary.* Both vocabularies are declared independently of the corpus they are applied to, but theirs is closed at constructs with a human-cognition analogue. Ours adds cards induced from benchmarks and cards hypothesised ahead of the corpus; 176 of the latter are demanded by no benchmark, which is what lets an untested capability appear as a gap rather than be absent from the map.
- *Resolution and inference.* Co-occurrence is analysed across the 14 domains only, with lift against independence and a floor of 20 papers. Those domains sit at our umbrella and coarse rungs, where coverage of pairs is nearly complete (119 of 120 umbrella pairs, 51 % of coarse pairs, 7.5 % of fine pairs are co-assigned by at least one benchmark). Lift also ignores that broad records carry many labels; a null that preserves each record's breadth and each label's frequency is the appropriate baseline.
- *What a label means.* Their label records that a paper studies a domain, graded strong (a central object of study) or weak (a supporting or contextual construct). The grade describes the paper's focus and is settled by model consensus, with ties resolved to weak; its reliability is not reported. Prevalence is given under both grades, but the subskill and co-occurrence analyses use strong labels alone, and no conclusion is re-checked with weak labels included. Our label states that a benchmark demands a card (Definition 9). Where the evidence is uncertain the uncertainty is kept rather than resolved: the intersection of the two raters' labels is the lower (sure) reading and their union the upper (possible) reading (Definition 15). Every completion between them is admissible, coverage is proved to lie between the two readings (Theorem 16), and the labeller is scored against both (Table 9).
- *What a count establishes.* Their paper reads low prevalence directly as neglect and co-occurrence as joint study, without stating what either licenses. The model of Section 2.1 states it: absence holds only within a label envelope, and co-assignment to a record does not establish joint exercise within an item (Theorem 10).

The comparison of their lift statistic with the exact-margin null on our matrix is in Section 3.

Two lessons from cognitive diagnosis. The attribute hierarchy method (Leighton et al., 2004) orders attributes by prerequisite. Our ladder instead records subsumption; the two relations need not have the same interpretation. Two methodological lessons carry over. First, identifying a Q-matrix from responses requires assumptions about the response model, item design and attribute distribution; observationally equivalent matrices may otherwise remain. Second, evidence-centred design (Mislevy et al., 2003) asks that the chain from an observed response to a claim about a latent attribute be written down as an explicit argument. Our setting differs in one respect. Cognitive diagnosis can check a Q-matrix against how examinees answered; benchmarks are not items put to a single population, and each is run on a different set of models, so nothing here plays that role.

The skill a subject actually uses. Item response theory has been used in NLP to estimate item difficulty and discrimination (Lalor et al., 2024). It infers ability under an assumed attribute structure, but a model may solve an item by a route its designer did not intend — a surface heuristic, a memorised answer, or leakage of test material into training (Chen et al., 2025). The response is then evidence about the shortcut and not the intended capability. Chollet (2019) states the general form: skill at a task is modulated by prior knowledge, so task performance is a poor proxy for the capability the task was built to probe. Response data therefore cannot validate a capability label without an assumption about the solving process. This is why our labels are statements about design, and why the release format of Section 6 asks authors to declare the structure rather than leave it to be inferred from scores.

Directionality in software testing. In metamorphic testing (Chen et al., 1998; Segura et al., 2016) and behavioural test suites (Ribeiro et al., 2020), a violated relation is informative about a failure, while a satisfied one establishes nothing about correctness. This is the same asymmetry as ours, in a different setting.

The labeller's bias. LLM judges are known to carry systematic preferences when ranking responses (Zheng et al., 2023; Li et al., 2026). The concern in our setting is specific to labelling artefacts: a judge that meets a thinly documented benchmark can supply capabilities rather than abstain. This is a failure mode to audit, not established by label breadth alone. A few very broad records can strongly influence coverage, so label breadth is reported as an influence diagnostic; whether a broad record's breadth is justified requires independent source review.

D CORPUS AND VOCABULARY DETAILS

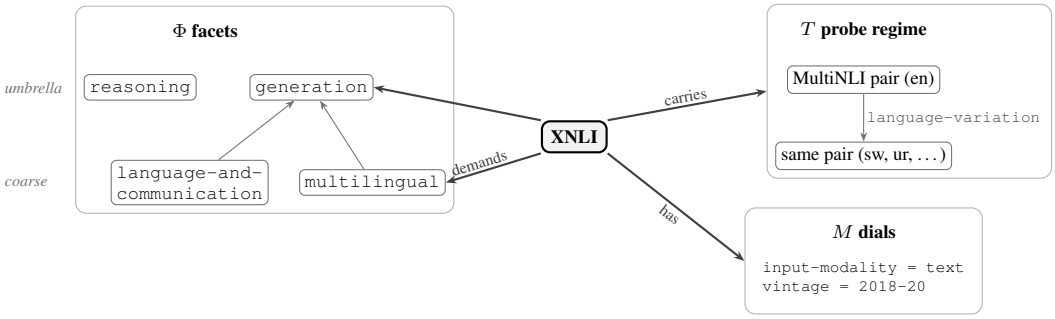


Figure 4: The three kinds of card, with XNLI's released labels (four of its six facets shown). A facet (Φ) names a capability the benchmark demands; facets are ordered by specificity (arrows point to broader cards), and only facets enter the coverage table. A probe regime (T) presents the same items differently with the answer unchanged: XNLI is MultiNLI, which is in the corpus and carries no probe regime, under `language-variation` (vote 0.82). A dial (M) records a property of the setup. Probe regimes and dials select benchmarks (Theorem 19); they are never demands.

The three kinds of card. Benchmark documentation mixes the evaluative target, the procedures used to vary or probe it, and properties of the setup, and a coverage table that counted all three

would count different things. The definitions of Section 2.1 are applied through contracts. A probe-regime assignment identifies the reference and the task contract held fixed (task rules, legal actions, resource constraints), the variation, the instances it applies to, and the relation between reference and resulting correct outcomes; answer equality is the invariant special case, and preserving the gold answer alone does not preserve the task. A dial reading is one of four states: a known value, an unknown value, inapplicable, or unresolved applicability; a missing field is unknown, not a value. Five recurring boundary cases are ruled explicitly. An adversarially constructed set has no reference instance and applies no probe regime; its construction is setup metadata. In a refusal benchmark the refusal is the facet (a disposition), the harm category describes the requests, and the harmful ability is not assigned. Selecting a curated subset transforms nothing and is setup metadata. Changing the evaluated system, e.g. by fine-tuning it, is configuration, not a probe. Translation is a probe regime per item, on the domain where the answer survives it. Changing a simulator's rules or budget is likewise a setup variant, which is why research under a compute budget is recorded by the dial `resource-budget` rather than by a tag. The facet labeller was run against facets only, the probe regimes were labelled by a separate vote over the 17 cards, and the dials are parsed from curated fields. Roles describe what an entry asserts; they do not assert that the underlying concepts are unrelated.

Imported frameworks. The 179 imported cards come from, among others, the O*NET work-activity taxonomy (Peterson et al., 2001), Cattell–Horn–Carroll (McGrew, 2009), Bloom's revised taxonomy (Anderson & Krathwohl, 2001), the SOLO taxonomy (Biggs & Collis, 1982), Spelke's core knowledge (Spelke & Kinzler, 2007), Grice's conversational maxims (Grice, 1975) and the NIST AI Risk Management Framework (National Institute of Standards and Technology, 2023).

Umbrella attachments. The umbrella rung is a sixteen-card cognitive vocabulary, and the ladder attaches everything to it, including capabilities it was not built to describe: `material-fabrication-and-assembly` reaches the top through `executive-functions` and `problem-solving`, and `psychomotor-abilities` through `executive-functions`. Each attachment is subject to semantic judgment.

E LABELLING PIPELINE DETAILS

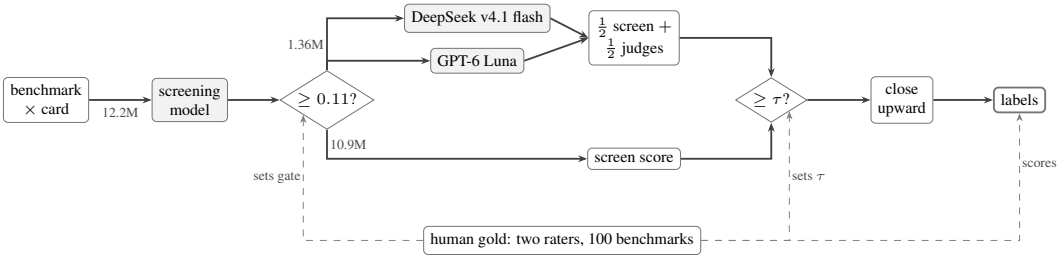


Figure 5: The labelling pipeline. Every (benchmark, card) pair is screened; pairs at or above the gate are re-judged by two judge models and scored by the vote, the rest keep the screening score. A card is assigned when the score reaches $\tau = 0.47$, and assignments are closed upward along the ladder. The human gold (dashed) sets the gate and the threshold and scores the result.

Design. The screening model, Jev (`typesafe/jev-1.13`, pinned; TypeSafe AI, 2026), sees thirteen substantive fields of a benchmark's card and, for each capability, its name and full definition, and answers one yes/no question per card — is this capability a load-bearing demand of the benchmark? — as a probability; 200 questions go in one request. Every pair it scores at or above the gate is repacked into blocks of at most 50 cards, 35,276 blocks per judge, and judged once by DeepSeek v4.1 flash (served by CoreWeave, synchronously) and once by GPT-6 Luna (served by OpenAI, through its batch interface). Hosts are pinned, with no fallback to another provider. DeepSeek looped on its hidden reasoning in 86 blocks; re-asking them with a larger token budget, and a lower reasoning effort for the last 18, answered every one, so every routed pair has both judges' scores.

The gate. Table 8 sweeps the gate. The shipped cut, 0.11, is the largest that keeps at least 95 % of the intersection-gold positives. A lower gate is not worth its cost: on the 25 benchmarks where both judges scored whole blocks, the pairs Jev scores in $[0.06, 0.11)$ hold 11 intersection positives among 367 negatives, and a vote near the shipped threshold rescues between one and two and a half of them while adding one to ten false positives (mean over two runs).

The vote. Table 9 compares ways of combining the three scores under nested leave-one-benchmark-out evaluation: for each gold benchmark, the threshold — and, for the fitted votes, the coefficients — are chosen on the other 99. Adding the judges is what matters; how they are weighted barely does. A sweep of the screening model's weight picks 0.6 with one judge and changes F_1 by $+0.005$ over equal weights, and a logistic regression learns near-equal coefficients. Intervals are benchmark bootstraps over the 100 held-out benchmarks. As a held-out check, the gate and the vote were designed on 25 pilot gold benchmarks; fitting the threshold, and the fitted votes' coefficients, on those 25 and scoring once on the other 75, which no design choice used, the shipped vote reaches F_1 0.597 $[0.552, 0.639]$ on raw labels against the intersection gold, within $+0.008$ $[-0.001, +0.018]$ of the equal three-way vote, and the fitted votes do no better.

Annotation conventions affect agreement. Scored as a third annotator on the 7,603 reviewed pairs, the labeller agrees with one rater at F_1 0.655 and with the other at 0.593, against 0.735 between the two humans. Some disagreement appears related to assignment depth. The raters have umbrella-to-fine assignment ratios of 0.71 and 2.32 (χ^2, $p = 5.6 \times 10^{-7}$). One ruling on depth would affect 289 contested rows, and eleven proposed subtree rules affect a further 472. These rule-based reconciliations are not new independent human judgments; the labeller must be rescored against any revised gold before comparing agreement. The measured correlations between apparent card vagueness and contested rate are below 0.09, which does not rule out wording effects. Forty-three cards are contested more often than jointly kept and warrant targeted review.

The earlier labeller. A first labelling of 11,529 benchmarks used a two-model cascade: a cheap gate model (`deepseek-v4-flash`; DeepSeek, 2026b) scored every card in 16 blocks of 50, an expensive judge (`glm-5.3-flash`; Z.ai, 2026) rescored the blocks the gate flagged, and the two scores were blended, 0.55 judge plus 0.45 gate, at a threshold of 0.39. No figure in this paper uses its labels. On the gold, the present vote ties it — every difference is inside its 95 % interval — and beats the screening model alone. The cascade's probe series is what shaped the present design, above all the finding that a maximum over two noisy scores biases upward on negatives, which is why the vote averages.

The earlier cascade's probe series. Every figure in Table 10 concerns the earlier cascade and is measured on the 100 two-rater gold benchmarks (7,603 labelled pairs over 83 families) or on a 22-benchmark subset of them, across a probe series run between 2026-09-06 and 2026-09-14 for a measured spend of $72.41. Intervals are bootstrap confidence intervals, clustered by benchmark family where the probe reports them. We set a resolution floor of 0.03 F_1 in advance: an effect below it is reported as null regardless of its point estimate.

Cost. The labelling of the 16,217 benchmarks cost $267.48: $92.73 for the screening model over all 12.2 million pairs, $158.58 for DeepSeek and $16.17 for GPT-6 Luna, whose batch interface halves the price, over the 1.36 million routed pairs. That is under two cents a benchmark, against the $500 cap the earlier cascade ran under for 11,529.

Provider variance. Identical prompts routed to different hosts of the same model do not return identical results. Six providers in the probe series returned incomplete outputs that were subsequently treated as missing judgments. Response completeness must be checked separately from syntactic parsing. Reconstructing the judge table from cache while filtering those hosts changes the measured AUC by 0.025, which is larger than most entries in Table 10. The final run therefore pins every host and allows no fallback to another provider. We also measured a drift of $+0.080$ F_1 $[+0.014, +0.160]$ in the same model over two months, which we cannot explain and did not control for. Host identity should be treated as part of the experimental condition when reproducing a model-labelled corpus.

F SUITE AUDIT AGAINST MATCHED SUITES

Given a suite, the audit reports which corpus-supported pairs (co-assigned by at least five corpus benchmarks) its members co-assign — the suite's *reach* — and which member witnesses each. Two bars are involved: five corpus witnesses for the denominator, one member for reach. Each member is replaced by a random non-member with the same card-count decile, primary input modality and era (2,000 draws; benchmarks whose modality is only a parser default form their own stratum, as in Section 4). A result is *robust* if the percentile criterion survives dropping any one member (500 draws each). The prespecified primary test is member overlap (mean pairwise Jaccard, Holm over the six suites within a rung); reach was also prespecified, with a per-suite percentile criterion that does not correct across suites.

No real suite passes the overlap test at either rung (AA at fine: Holm 0.11); the Chinese portfolio passes at both rungs (fine 0.03) and is robust to dropping any member, and the multilingual portfolio passes at coarse. Under an earlier matching key that put the parser-default benchmarks in the text stratum, AA and the HELM list also met the reach and leave-one-out criteria and AA's fine overlap passed (Holm 0.02); with the defaulted benchmarks in their own stratum they stay below the matched null but no longer meet the leave-one-out criterion. At the fine rung reach lies inside the null for every real suite. Across a grid of five label policies and two rungs (20,000 draws per cell), no suite's reach deficit is robust; the Chinese portfolio is below the matched median in 9 of 10 cells and significant in 5, AA in 1, other real suites in none. The matched null conditions on breadth, modality and era; reach measures reach into this vocabulary, not adequacy, and co-assignment is an upper reading of joint testing.

Members. Artificial Analysis Intelligence Index: AA-Briefcase, GDPval-AA, AutomationBench-AA, Terminal-Bench, SciCode, Humanity's Last Exam, GDP-PDF, CritPt, AA-Omniscience, AA-LCR v1.1. HELM Capabilities list: MMLU-Pro, GPQA, IFEval, WildBench, Omni-MATH. Open LLM Leaderboard v1: ARC, HellaSwag, MMLU, TruthfulQA, Winogrande, GSM8K; v2: IFEval, BBH, MATH, GPQA, MuSR, MMLU-Pro. Chinese portfolio (author-constructed): C-Eval, CMMLU, AGIEval, GAOKAO-Bench, SuperCLUE, AlignBench. Multilingual portfolio (author-constructed): MGSM, XNLI, Belebele, Global-MMLU, XCOPA, TyDi QA, MMMLU.

Table 12 gives the reach of four declared suites as shares of all corpus-supported pairs, without a matched comparator. A single member often carries a suite: MMLU witnesses 48 % of what Open LLM v1 reaches at coarse, MuSR 55 % for v2 and Omni-MATH 65.5 % for the HELM list. As an illustration of selection, five greedy additions to Open LLM v2 from a breadth-capped pool gain 140 fine supported pairs against 21.3 ± 12.9 for random additions. This optimises the same label objective and does not show better model diagnosis: the picks include robotics and segmentation benchmarks unsuitable for a text-model suite, and a count of five is not a cost budget. A border-completion rule did not beat ordinary multicover (Table 15).

G RESOLUTION AND CONCENTRATION DETAILS

Bar sweep without speculative members. Share of co-assigned pairs reaching the bar, coarse / fine: bar 2, 87.8 / 68.9 %; bar 3, 80.1 / 57.0 %; bar 5, 70.7 / 42.8 %; bar 10, 58.2 / 28.6 %; bar 20, 47.1 / 18.6 %. The majority reversal holds at bars 5 and 10. Fine all-pair coverage at bar five rises from 2.7 % to 7.6 % when speculative members are excluded: the denominator changed, not evaluation practice. Speculative cards are also labelled less reliably (machine precision 0.18 against 0.55 on the rest).

Taxonomy sensitivity. None of 60 random same-shape groupings of the fine cards falls below 50 % demanded-pair coverage (range 55–62 %; the ladder gives 46.5 %). This is a different quantity from the share of co-assigned pairs reaching a bar, and is not a validation of the resolution result.

Further concentration statistics. The top 1 % of coarse pairs hold 25.2 % of pair support against 23.6 % under the Curveball null (40 draws); on Papers-with-Code tags, 9.2 % against 5.9 %. The corpus fills 46.8 % of the Kruskal–Katona cap for pairs and 42.8 % for triples at the coarse rung. Nesting (fewer maximal profiles than the null) does not survive both union label-noise arms and is

only supporting evidence. After stratifying by construction profile, 285 of 521 subject–card associations (55 %) no longer meet the selection criterion, against about 10 % under random strata of the same sizes.

H DEFICIT QUEUE DETAILS

Sets. Under the global fit 512 pairs meet the three-threshold rule and within input-modality strata 274; 198 meet both. Benchmarks whose input modality is only a parser default (2,048) form their own stratum; placing them in the text stratum instead gives 335 and 250, dropping them from the within-strata fit 246 and 181.

Policy robustness. Share of the candidates that remain candidates under each alternative policy (global 512 / within 274 / both 198): looser threshold fitted to the union gold 76 / 68 / 67 %; human-standard model 81 / 80 / 84 %; its conjunction with the labels of record 80 / 81 / 84 %; all three 60 / 58 / 58 % (305 / 158 / 115).

Exact-margin checks. Curveball within the same strata, 200 draws: relational reasoning × video 12 against 140.3 ± 7.9; code generation × video 13 against 126.0 ± 8.1; factual knowledge × video 28 against 161.7 ± 9.6. Multi-turn coherence × quantitative knowledge (33 against 240 expected globally) is a global candidate only.

Language pairs. Thirteen of the 115 involve CJK or Arabic script discipline, low-resource multilingual pragmatics or morphologically rich language fluency, paired with agentic, code, grounding, mathematical or quantitative cards.

Triples. Among 48,147 coarse triples whose pairs are all supported at bar five ($E \geq 10$), 93 (0.19 % $[0.07, 0.39]$) are co-assigned at most a quarter as often as a pair-conditioned model expects; three-way excesses outnumber deficits by 2.5 points $[0.7, 4.5]$.

I CONSTRUCTION CASE: PROTOCOL AND DISCLOSURES

Tests. Exact sign tests on discordant held-out items (control right only against full right only), pooled over the four mid-tier models and Holm-adjusted: facts given 91 against 14, $p \approx 10^{-13}$; belief only 91 against 10; quantitative only 83 against 18 (both $p < 10^{-9}$); retrieval only 59 against 31, $p = 0.029$; the pairs that keep retrieval 41 against 50 and 52 against 40, $p = 1$. Per model, the facts-given and belief-only contrasts hold for all four, the quantitative-only contrast for two, and the retrieval-only contrast for none (Luna and GLM at Holm $p \approx 0.06$); GLM's output cap, which counts capped answers as wrong, may inflate its full-item difficulty. Pooled over development and held-out items the same comparisons hold at Holm $p < 0.01$, except the retrieval pairs ($p = 1$); an item-clustered sign-flip check leaves these conclusions unchanged. The tests treat model–item pairs as units and are restricted to the four named models; the two ceiling models cannot show a full-item difficulty and are not pooled. Score formats differ across controls (probability, list, direction), so accuracy contrasts are not pure effects of removing one capability; the facts-given pair keeps the quantitative target and is the cleanest contrast. About half the model–item pairs with all three single cards right are still failed in full (63 of 127). No cost per added skill was established on the skill ladder (-4.0 points per skill, $[-8.5, +0.5]$).

Protocol and disclosures. The release has 28 development, 190 skill-ladder and 48 held-out items, each in seven variants: the full question and six demand-removal controls, the three pairs of cards and the three single cards. Unlike AutoBencher (Li et al., 2025), the description comes from the detected gap rather than a proposing model. The package `reproducible/cell67` recomputes every number in §5 except the manual screening of the top 80 cells and the reading of the four labelled carriers. The four mid-tier models have all six controls on both splits (Table 14); the sign tests above are on held-out items, each comparison paired on the same world, and the check pooled over development and held-out items is reported beside them. The two stronger models, DeepSeek-v4.1 flash and Gemini 3.8 Flash, also have all six controls on held-out, at ceiling (92–100 %) like their full scores, and are not pooled, since a model at or near 100 % on the full item cannot show it

to be harder than a control. Empty answers from provider errors were re-run; GLM-5.3 flash hit its output cap on many items, and answers at the cap count as wrong, so its counts are lower bounds. Six models were evaluated, all on the full held-out split with every control (Figure 3). Two further models were run on development items only, entered no test and are not reported; one is the renderer, which should not be scored on its own output. The 48 held-out items and their keys ship in full so that every number can be recomputed; the hash-only held-out split of Figure 6 is a proposal and is not applied here. The renderer (Qwen 3.8 Flash) and the read-back validator (DeepSeek-v3.2) are language models; the validator is also an evaluated model but only ever saw single episodes during validation. Template text instead of prose on the same worlds is easier ($p = 0.004$ on both splits). Passed through the census labeller, all ten write-ups of the benchmark land on the three target cards; they were written knowing the target, so this shows robustness to wording, not independent confirmation.

J TESTS THAT DID NOT SUPPORT A CLAIM

We report the registered or pre-specified tests whose outcome did not support the claim they were designed for. None bears on the descriptive results of Section 3; together they bound what the atlas has been shown to do.

K REPRODUCIBILITY

We release the vocabulary, the facet and probe-regime labels, the dial values, the human gold, the Lean development and the analysis code. A single script regenerates every table and figure of the paper from the frozen labels. Before any analysis it checks the inputs against their recorded hashes and re-checks the structural hypotheses the theorems use: an antisymmetric order, antichain rungs, levels strictly monotone along the order, and every card reaching the umbrella rung. The Lean development builds with an audit of the axioms it uses, and a check keeps the card definitions printed in the paper identical to their source and to the Lean types. Recomputing the statistics from the frozen labels is distinct from repeating the labelling run, which calls external model providers and is stochastic; Appendix E records the models, hosts, gate and threshold of the run we report. Code and data are provided as supplementary material.

L THE PROPOSED RELEASE FORMAT

The four parts. *A construct declaration*, at benchmark level: which capabilities the benchmark claims to test, at which rung, against which vocabulary version. It supports checks of whether every claimed capability is demanded by enough items, which unclaimed capabilities the items also demand, and whether every claimed capability has an item that demands it alone. The last is a completeness condition in some cognitive diagnosis models; identifiability also depends on the response model and attribute structure (Liu et al., 2012).

An item table: per item, the capabilities it demands, whether it demands them jointly or separately, its answer hash, the dial settings under which it is presented, its scoring rule, and, for an item derived from another by an answer-preserving transformation, a link to its parent and its own capability vector (Figure 6). The variant's vector must be declared rather than inherited, because a transformation that preserves the answer key can still destroy the test; this is a theorem in our development.

A response table, in long format: one row per (model, item, attempt), with the decoding configuration, the date and raw scores. This is the data over which "tests" is defined; a benchmark that publishes only a mean has published a quantity from which it cannot be recovered.

A declared measurement model, optionally: whether multi-capability items are meant to be compensatory or conjunctive, with loadings constrained to the declared capabilities and to a fixed orientation. Scores alone do not distinguish a capability from its complement, so the orientation has to be declared by the people who wrote the items.

What it buys. We read every co-assignment as joint, which credits the most combination coverage; a declared `joint: false` can only remove coverage, never add it. With the orientation

```
{ "item_id":     "cxr-0413",
  "q":           ["chart-and-diagram-interpretation",
                  "numerical-precision"],   // capabilities demanded
  "joint":       true,            // demanded together, not separately
  "vocabulary":  "...",         // release the names resolve against
  "answer_hash": "sha256:...",    // checks a variant against its parent
  "regime":      null,            // transformation applied, or null
  "parent_item": null,            // the item this one transforms
  "dials":       {"input-modality": "image", "output-format": "number"},
  "scoring":     {"type": "binary"},
  "split":       "public" }       // held-out items publish hashes only
```

Figure 6: An illustrative item-table row in JSONC notation. A common schema makes capability declarations, joint-demand claims and item lineage explicit. Hashes can support exact-match integrity checks when the corresponding candidate content and hashing convention are available; they do not generally detect training contamination or paraphrased exposure.

declared, an item that discriminates negatively flags a mismatch to investigate in the capability declaration, response model or solving process; it does not uniquely identify a wrong matrix entry.

What it costs. Authoring Q-vectors by hand is expensive, and proposing them with an automatic assigner inherits that assigner's biases, which then have to be disclosed. Items and responses leak the test, which is what the held-out split is for. Closed models cannot be back-filled, so early adoption produces a response table with a gap where the frontier models should be. And the format does not make a capability claim causal: it supports score separation between models that differ on a capability, which is weaker than a claim that the capability produces the score.

Table 5: Statements of Appendix A and their Lean 4 declarations.

statement	declarations (module)
Typing	`Setup`, `Dial`, `Perturbation` (probe regime), `Perturbation.comp`, `Capability`, `Tests`, `Invariant`, `perturbation_may_change_score`, `perturbation_may_destroy_test` (Typing)
Prop. 6	`tests_not_determined_by_artifact` (Typing)
Prop. 7	`capability_not_identifiable_from_scores` (Typing)
Card space	`Card`, `Card.Le`, `Card.kind_eq_of_le`, `Card.dial_le_iff_eq`, `Card.phi_le_iff` (Typing)
Demand order, Def. 8	`Order.Refines`, `Order.refines_iff_subset_of_discrete` (Order.Refinement); `Demands`, `demands_of_subset`, `demands_singleton`, `demands_iff_subset_of_discrete` (Demand)
Coverage and gaps, Def. 9	`Corpus`, `CoveredClosed`, `CoveredNative`, `coveredClosed_not_coveredNative`, `Complete`, `Gap`, `CoveredClosed.mono`, `Gap.of_demands`, `MinimalGap` (Demand)
Thm. 10	`CoveredUnder`, `CreditsNoMore`, `coveredClosed_of_coveredUnder`, `gapUnder_of_gap`, `gapUnder_not_gap` (Demand)
Def. 11, Prop. 12	`WorkingCut`, `project`, `project_pairwise_incomparable`, `project_mono`, `project_eq_of_upward_closed`, `CoveredAt`, `coveredClosed_of_coveredAt`, `GapAt` (Resolution)
Rungs, Prop. 13	`Stratification`, `Stratification.rungCut`, `Stratification.lvl_eq_of_subset_rungCut`, `Stratification.biUnion_rung_eq` (Resolution); `Rooted`, `NoRungSkip`, `lvl_succ_of_covBy`, `exists_ancestor_at_level`, `project_rung_nonempty`, `project_eq_empty_of_lvl_lt` (Rootedness)
Thm. 14	`Finer`, `Resolves`, `Unambiguous`, `coveredAt_singleton_of_finer`, `gapAt_singleton_of_finer`, `card_gapCards_le_of_finer` (Resolution); `gap_count_not_grain_invariant`, `no_confirmedGap_in_fineCut` (GrainInvariance)
Gap status	`Unresolved`, `ConfirmedGap`, `status_trichotomy`, `confirmedGap_mono_le`, `not_coveredClosed_of_le_of_confirmedGap` (GapStatus)
Def. 15, Thm. 16	`Obs`, `PartialCorpus`, `obs`, `lower`, `upper`, `IsCompletion`, `IsRefinement`, `isRefinement_not_isCompletion`, `coveredClosed_refinement_of_lower`, `coveredClosed_upper_of_refinement`, `isCompletion_sure`, `isCompletion_possible`, `countCovered_completion_bounds`, `countCovered_powersetCard_bounds`, `undetermined_iff_exists_completions_disagree`, `CertainGap`, `certainGap_of_demands`, `undetermined_loosen_of_unresolved` (PartialObservation)
Two annotators	`ofTwoAnnotators`, `lower_ofTwoAnnotators`, `upper_ofTwoAnnotators`, `obs_ofTwoAnnotators_eq_unknown_iff` (PartialObservation)
Prop. 17	`coveredAtOrder`, `coveredAtOrder_eq_biUnion`, `card_coveredAtOrder_le`, `coveredAtOrder_eq_empty_of_lt`, `coveredAtOrder_shadow_subset`, `shadow_iterate_coveredAtOrder_subset` (Combination)
Thm. 18	`cascadeTop`, `cascadeShadow`, `cascadeUpper`, `cascadeShadow_le_iff_le_cascadeUpper` (Cascade); `choose_le_card_shadow_iterate` (KruskalKatona); `colexRank`, `cascadeShadow_le_card_shadow'`, `exists_card_eq_cascadeUpper_of_le` (CascadeFamily); `cascadeShadow_card_coveredAtOrder_le`, `card_coveredAtOrder_le_cascadeUpper`, `card_coveredAtOrder_lt_of_sub_lt` (Combination)
Filled and empty regions	`Hyperbox.Filled`, `Hyperbox.Empty`, `Hyperbox.Filled.mono`, `Hyperbox.Empty.mono`, `Hyperbox.filled_one_iff`, `Hyperbox.filled_iff_exists_maximal`, `Hyperbox.empty_iff_exists_minimal`, `Hyperbox.not_filled_of_contains` (Hyperbox)
Items	`ItemCorpus.toBench`, `ItemCorpus.bench_covers_no_item_does`, `ItemCorpus.gapUnder_of_items_below`, `ItemCorpus.gapUnder_itemwise_of_gap`, `ItemCorpus.creditsNoMore_of_items_below`, `ItemCorpus.gap_given_toItems_of_gap`, `ItemCorpus.narrowUnder_any_not_items`, `ItemCorpus.narrowUnder_toItems_of_all` (Items)
Observations	`Observations.toMultiDial`, `Observations.narrowUnder_of_certified` (Observation)
Probe applications	`ProbeApplication`, `Perturbation.toApplication`, `ProbeApplication.comp`, `vacuous_application`, `ProbeApplication.NonVacuous`, `ProbeApplication.AssumptionSatisfied`, `valueSubstitution_not_answer_only` (ProbeApplication)
Cascade values	`Examples.CascadeV13` (Examples.CascadeV13)
Further scope results	`Corpus.withDials`, `coveredClosed_withDials_iff`, `narrowUnder_two_values` (Conditioning); `confirmedGap_iff_maximal` (GapStatus); `not_confirmedGap_of_rooted` (Rootedness); `confirmedGap_count_not_grain_invariant` (GrainInvariance); `lower_ofTwoAnnotators_grain_mismatch`, `refinement_below_label_escapes_upper`, `coveredClosed_upper_of_downClosed` (PartialObservation)
Trades	`Trade`, `Trade.card_right`, `swap_card`, `swap_column_sum` (Randomization)
Thm. 19	`Corpus.restrict`, `coveredClosed_restrict_iff`, `coveredAt_restrict_iff`, `CoveredClosed.of_restrict`, `CoveredAt.of_restrict`, `Gap.restrict`, `CoveredClosed.restrict_mono`, `MultiDial`, `Corpus.given`, `coveredClosed_given_iff`, `coveredClosed_iff_exists_given`, `covered_without_any_given`, `NarrowUnder`, `narrowUnder_iff` (Conditioning)

Table 6: The four rungs of the facet vocabulary. *Reaching*: benchmarks of the 16,217 with at least one facet on the rung after upward closure. Provenance: imported / corpus-induced / hypothesised (five coarse cards have no recorded source). The atom rung holds rare specialised capabilities; its 27 cards sit below 17 of the 531 fine cards, so it is not a full refinement of the fine rung.

rung	cards	reaching	provenance
umbrella	16	16,214	13 / 0 / 3
coarse	180	16,211	66 / 21 / 88
fine	531	14,471	100 / 76 / 355
atom	27	730	0 / 13 / 14

Table 7: The three kinds of card, with familiar and unusual examples of each, over the 16,217 benchmarks. *Benchmarks* counts, for a facet, records demanding it at $\tau = 0.47$ (after upward closure); for a probe regime, records carrying it at the probe-regime labels' own cut ($\tau = 0.1125$), labels that over-assign (precision 0.30 on their gold set); for a dial, records with a known value, which is a parsed property of the setup and not a demand. The facet column spans four orders of magnitude of support, down to a hypothesised card no benchmark demands.

card	rung	benchmarks	what it records
Evaluative facet (754) — the evaluative target			
`reasoning`	umbrella	12,839	drawing valid conclusions by logical inference
`stem-knowledge`	coarse	7,219	science, technology and engineering content
`maxim-of-manner`	fine	431	presenting the required output clearly, in Grice's sense
`goal-preservation-under-self-modification`	atom	1	holding goals fixed across fine-tuning, distillation or capability extension
`corvid-caching-and-future-planning`	fine	0	scrub-jay style mental time travel: caching, spoilage, re-caching when watched
Probe regime (17, flat) — a controlled probe variation			
`robustness`	—	4,438	surface and form variations of an instance with the answer unchanged: paraphrase, typos, reordering
`consistency-under-resampling`	—	2,498	the identical instance administered again: another seed, session or run
`language-variation`	—	1,415	the instance translated or rendered in another language, where the answer survives it
`decision-distorting-pressure`	—	493	improper pressure added to a judgement, with the correct outcome unchanged
Metadata dial (14) — a property of the evaluation setup			
`input-modality`	—	16,217	which input channels the benchmark presents
`scoring-protocol`	—	15,521	how a response is turned into a score
`vintage`	—	14,834	publication era, carrying contamination and saturation risk
`resource-budget`	—	10	the budget the evaluation declares for a task or run, with amount and unit

Table 8: The gate: the share of gold positives it keeps and the pairs it routes to the judges, over all 12,227,618 pairs of the corpus.

gate	kept, intersection	kept, union	routed pairs	share
0.05	99.7 %	97.5 %	3,233,779	26.4 %
0.08	98.3 %	93.2 %	1,956,781	16.0 %
0.11	**95.9 %**	**88.0 %**	**1,364,118**	**11.2 %**
0.15	91.9 %	80.6 %	962,549	7.9 %
0.20	87.1 %	73.1 %	692,084	5.7 %

Table 9: The labeller against the human gold, nested leave-one-benchmark-out over the 100 gold benchmarks (7,598 pairs). Intersection counts a pair positive if both raters kept it, union if either did.

			intersection		union
vote on routed pairs	P	R	F_1 [95 %]	AUC	F_1 [95 %]
screening model alone	0.581	0.606	0.593 [0.554, 0.623]	0.861	0.634 [0.599, 0.663]
mean of screening model and Luna	0.604	0.676	0.638 [0.601, 0.668]	0.878	0.648 [0.612, 0.678]
mean of all three	0.618	0.647	0.632 [0.594, 0.662]	0.883	0.654 [0.619, 0.682]
$\frac{1}{2}$ **screening** + $\frac{1}{2}$ **judges' mean**	**0.612**	**0.690**	**0.649** [0.611, 0.679]	**0.884**	**0.658** [0.624, 0.685]
logistic regression, raw scores	0.609	0.687	0.646 [0.609, 0.675]	0.880	0.654 [0.621, 0.682]
logistic regression, logits	0.608	0.694	0.648 [0.611, 0.677]	0.879	0.644 [0.609, 0.673]

Table 10: What was tried on the earlier cascade. The first block is what it shipped with; the second is what did not survive a control. Effects are against the configuration the probe was testing against, so they do not add.

change	measured effect	verdict
Kept		
max fusion → weighted blend	+0.093 F_1 [+0.069, +0.119]	shipped
keep gate score on unjudged blocks	+0.048 AUC	shipped
second model in the gate, not the panel	−0.005 AUC [−0.015, +0.004], −37% cost	shipped, then retired on price
block size 50 vs 25 / 100	+0.078 / +0.103 F_1	shipped
blend vs strong model alone	+0.013 [−0.002, +0.028]	kept, but no evidence it helps
Null		
learned per-benchmark threshold	+0.005 [−0.009, +0.018]	null
per-rung thresholds	+0.001 ± 0.003	null
stacked scorer (logistic / GBT)	+0.013 [−0.002, +0.028]	null
frozen per-block offsets	F_1 +0.011 [−0.001, +0.026]	null on F_1
score-disagreement abstention	fails at every budget	null
discrete rubric instead of a score	AUC +0.010 [−0.016, +0.034]	null
base-rate anchoring	ECE −62%, no F_1 gain	superseded by isotonic
provider pinning	+0.015 F_1, CI spans zero	null
frontier judge at 24× the price	indistinguishable	null
`json_object` vs `json_schema`	−0.068 F_1 on the strong model	format-dependent
per-benchmark z-score / rank normalisation	0.921 → 0.906 / 0.898 AUC	harmful
lexical retrieval to shortlist cards	recall 1.0 needs 730 of 751 cards	no usable shortlist

Table 11: Coarse-rung reach against breadth-, modality- and era-matched suites. *pct*: share of matched suites with lower reach; *matched*: matched median; p: two-sided; *overlap Holm*: the primary overlap test. The HELM list is MMLU-Pro, GPQA, IFEval, WildBench and Omni-MATH, not all of HELM; the Chinese and multilingual portfolios are author-constructed positive controls.

suite (members)	reach	matched	pct	p	overlap Holm	robust
AA Intelligence Index (10)	643	909	0.004	0.009	0.184	no
HELM Capabilities list (5)	139	207	0.007	0.016	0.184	no
Open LLM Leaderboard v2 (6)	165	227	0.013	0.029	0.184	no
Open LLM Leaderboard v1 (6)	138	170	0.035	0.079	0.184	—
Chinese portfolio (6)	177	309	0.001	0.003	0.012	yes
multilingual portfolio (7)	121	159	0.013	0.036	0.020	no

Table 12: Unmatched suite reach (exploratory): corpus-supported pairs (at least five corpus witnesses) co-assigned by at least one member. Open LLM Leaderboard v1: ARC, HellaSwag, MMLU, TruthfulQA, Winogrande, GSM8K; v2: IFEval, BBH, MATH, GPQA, MuSR, MMLU-Pro; the HELM Capabilities list used here: MMLU-Pro, GPQA, IFEval, WildBench, Omni-MATH (not all of HELM). The ten-benchmark portfolio (HLE, GPQA, SWE-bench, Terminal-Bench, BrowseComp, MMMU, ARC-AGI, FrontierMath, SimpleQA, LiveCodeBench) is illustrative, not a model card.

suite	coarse (of 5,374)	fine (of 3,796)
Open LLM Leaderboard v1 (6)	138 (2.6 %)	9 (0.2 %)
Open LLM Leaderboard v2 (6)	165 (3.1 %)	18 (0.5 %)
HELM Capabilities list (5)	139 (2.6 %)	32 (0.8 %)
illustrative portfolio (10)	618 (11.5 %)	208 (5.5 %)

Table 13: Coarse pairs supported at a bar of five whose supporting benchmarks share one input modality value ($\tau = 0.47$). *Determined*: pairs whose confinement is decided, because either the supporters that state a modality already share none, or every supporter states one; a pair whose stating supporters share a value but which has a silent supporter is undetermined (Theorem 19). The primary variant treats a modality filled in by the parser default as unknown; the released dials treat it as a value. The two variants are over different pair sets. The permutation null uses 200 draws (40 in the third row); the modality-mix null uses 10 draws in every row, so the second and third rows are the same variant with different random streams. Prespecified.

variant	determined	confined	permutation	modality mix (z)
default unknown (primary)	4,469	289	9	457 (-6.9)
released dials	5,374	1,135	62	1,308 (-9.5)
released dials, 40-draw null	5,374	1,135	55	1,321 (-6.3)

Table 14: Construction case: items solved per model on the full question and its demand-removal controls, held-out split (of 48). *B*: belief-updating, *Q*: quantitative reasoning, *R*: retrieval-in-context; B+Q is the facts-given pair. GLM-5.3 flash hit a 32k output cap, so its counts are lower bounds.

model	full	B+Q (facts given)	R+Q	R+B	R
DeepSeek-v3.2	25	46	21	21	23
GPT-5.6 Luna	25	40	28	33	38
Qwen 3.6	21	42	18	21	24
GLM-5.3 flash	21	41	16	29	35
DeepSeek-v4.1 flash	46	47	45	46	44
Gemini 3.8 Flash	48	47	47	48	48

Table 15: Negative and null outcomes (retrospective and observational unless stated).

question	measured	verdict
Does combination rarity predict slower model progress on Papers-with-Code leaderboards?	$-0.086\ [-0.170, +0.002]$	null
Are rarely co-demanded abilities harder for LLMs on ADeLe items?	primary 0 of 15; held-out gain $\leq$ 0.0005 nats	not supported
Do capabilities differ in saturation on Ott et al.'s SOTA series?	$p = 0.58$; $\rho = -0.17$	not supported
Do bridge cards improve temporal-holdout forecasts of which pairs fill?	no gain over the pair graph	refuted
Is the multilingual action deficit testable at this corpus size?	direction confirmed, too rare	not supported as registered
Does a border-completion rule beat ordinary multicover for acquisition?	loses by 2–6 % at every budget	refuted
Is the AA Intelligence Index under-covering any capability, after correction?	no corrected test passes	not supported
Does facet overlap predict residual score correlation between benchmarks beyond setup dials?	$+0.12$ ($p = 0.046$); $+0.11$ after the dials ($p = 0.064$)	not supported
Are benchmarks that fill gaps more informative about models?	$\rho = -0.03\ [-0.31, +0.26]$, 42 benchmarks	registered null supported
Do atlas-chosen pairs (benchmarks to 2022) get built later more often than pair-gap picks?	share built 0.67 $[0.50, 0.83]$ vs 0.60 (random 0.30)	no gain over the baseline