SNAP: ERROR BARS ON BENCHMARK AVERAGES CAN MISS RUN COVARIANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Error bars on a language model's benchmark average usually sum per-benchmark variances and ignore the covariance between benchmarks across training runs. On 375 DataDecide runs, the standard deviation of the ten-benchmark average in per-byte margin is 1.244 $[1.143,\ 1.338]$ times its value under independence, while accuracy gives 1.078 with an interval that includes one. A margin comparison that assumes independence at a nominal 0.05 therefore has a true size near 0.11. That covariance sits within scoring formats, where BoolQ, alone in its format, carries 68% of the margin covariance trace. SNAP, a cross-half estimator, also gives per-benchmark run variances net of item noise, which the replicate standard deviation can't separate when item noise is large. A correction for the covariance changes few of DataDecide's recipe orderings. A pre-specified test on four held-out tasks fails at 1.217, with a lower limit below one. A second pre-specified rule also fails, since nine seeds at five PolyPythias sizes give margin inflation of 1.302 with an interval $[0.921,\ 1.595]$ that includes one. A disjoint item bank leaves the source of that inflation undecided. All other analyses in the paper are exploratory. We recommend that authors report the replicate standard deviation of the battery average with a benchmark-removal check, and that standard deviation needs no item split and gives an inflation of 1.237 here.

1 INTRODUCTION

A benchmark average can change across training runs even when the evaluation items stay fixed. If two runs of one configuration differ on that average, how much of the gap should an error bar expect? For an equally weighted average of K benchmarks, independent run deviations give variance $K^{-2}\,\mathrm{tr}\,\Sigma_E$, where Σ_E is their cross-benchmark covariance matrix. In other words, the naive independence assumption drops the off-diagonal terms and weakens a comparison's error control.

When the true standard deviation is Λ times the independent one, a two-sided comparison at a nominal 0.05 has size $2\{1 - \Phi(1.96/\Lambda)\}$ with Φ the standard normal distribution function. That size is 0.115 at the margin estimate and 0.069 at the accuracy one, and our simulation of two independently trained configurations gives 0.113 (Appendix C). Since the correction changes few calls on DataDecide's recipe comparisons, the understatement shows mainly in the error bar of one model's battery average and in planning replicate runs.

Figure 1 shows where that covariance sits in the DataDecide battery. We estimate item-general covariance from two disjoint item halves per benchmark, where we centre both half scores across each configuration's replicate runs and take their cross-half products. Under zero-mean item noise and conditional independence across halves, these products estimate run covariance with its diagonal. We adapt the repeated-measurement construction of quantitative genetics and call it SNAP, Seed Noise Across Phenotypes (Appendix A).

The empirical study uses DataDecide (Magnusson et al., 2025), with 25 data recipes and five of its model sizes, each with three released replicates, and those 125 configurations supply 375 runs across ten benchmarks, scored at the largest checkpoint step each configuration's replicates share. The primary analysis reads released item scores without model inference and compares per-byte log-likelihood margins with accuracy. Margins carry more seed signal, since their per-benchmark reliability (i.e., the run variance's share of the replicate variance of one benchmark half) has a mean

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.

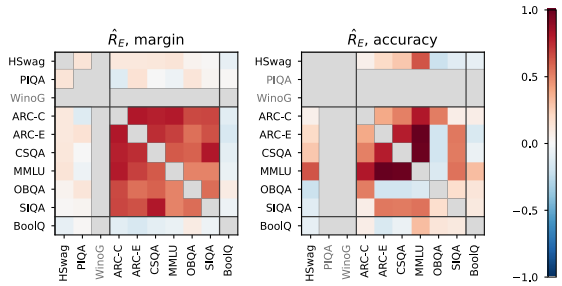


Figure 1. Cross-half correlation estimates of run deviations over the 125 DataDecide configurations, per-byte margins left and accuracy right, with undefined values in grey. Margin correlations average 0.624 within scoring formats and 0.007 across them (Section 4.1).

of 0.652 against 0.324 for accuracy (Section 4.1). We therefore treat margins as the primary scale and accuracy as a bound.

Margin inflation Λ, the ratio of the average's standard deviation to its independent value, is 1.244 $[1.143, 1.338]$. Accuracy gives 1.078 $[0.993, 1.157]$ at the scored checkpoint, an interval that includes one (Table 1).

Since we built the estimator on these ten benchmarks, we pre-specified a test on SciQ (Welbl et al., 2017), MedMCQA (Pal et al., 2022), and DROP (Dua et al., 2019) and CoQA (Reddy et al., 2019) in multiple-choice form, on the same checkpoints at 530M, 750M, and 1B. Its rule requires the lower interval limit of held-out margin inflation to exceed one. The test fails at its full scope of 225 runs, where held-out inflation is 1.217 $[0.872, 1.498]$ against 1.240 $[1.054, 1.397]$ for the original ten on the same runs (Section 4.2).

A single model family whose replicates mix default and auxiliary runs on different schedules leaves open whether the inflation belongs to DataDecide's design. We therefore score all 45 PolyPythias runs at five sizes, where each of nine seeds reruns one recipe, and pair that battery with a disjoint bank of 6,808 items built with the OLMES harness (Gu et al., 2025) at a pinned commit. Two pre-specified rules read the results, and in simulation the first passes in 0.259 of replicates at a true 1.244 and the second returns undecided in 0.761. Section 4.3 reports that margin inflation there is 1.302 with interval $[0.921, 1.595]$, which fails the first rule, and that the shared-item explanation is undecided.

This paper makes four contributions to the measurement of run covariance. (i) We define the item-general run covariance that cross-half products estimate, identified up to an item component shared across benchmarks and halves. (ii) The replicate standard deviation of the battery average gives the uncertainty of a rerun on the released items and is what we recommend that authors report. The split adds per-benchmark run variances net of item noise and a redrawn-bank target, untested on DataDecide, and its value departs from the replicate one when item noise is large, at 1.558 against 1.285 for accuracy without BoolQ (Section 2). (iii) Algorithm 1 gives the measurement recipe, with near-nominal intervals in simulation unless the runs of one model size share a run effect or item noise reaches the top of our swept range, and it tells a planner that, with generic item noise, three runs per configuration resolve margin inflation while accuracy needs eight. (iv) We measure run covariance across the 375 released runs and find it concentrated within scoring formats, where every recipe, recipe-pair and benchmark deletion and four of five size-band deletions leave a margin interval that excludes one.

2 Estimand and Estimator

2.1 Runs and Score Scales

Formally, let c index configurations, r index their R replicate runs, and j index benchmarks, where a configuration fixes a data recipe, model size, and nominal hyperparameters, and Table 3 collects the symbols. We call the set of benchmarks the battery and the set of items the bank. DataDecide's seed controls initialisation and data order together. Each configuration has one default and two auxiliary runs, and below 1B the auxiliary runs stop at 25 percent of the 1B compute budget. That schedule

difference adds a batch component beyond the seed. For item i, define the per-byte margin and corresponding correctness indicator by

$$m_{crji} = \frac{\ell(\text{gold}_i)}{\text{bytes}(\text{gold}_i)} - \max_{v \neq \text{gold}_i} \frac{\ell(v_i)}{\text{bytes}(v_i)}, \qquad a_{crji} = \mathbf{1}\{m_{crji} > 0\}. \tag{1}$$

Here ℓ is the summed answer log-likelihood and v ranges over alternatives, with a zero margin treated as incorrect. We average items per benchmark, with MMLU macro-averaged over its 57 subjects, and we denote the margin and accuracy scores by y^{marg} and y^{acc} and call each a trait.

For each half $H \in \{A, B\}$, the measurement model is

$$y^H_{crj} = \mu^H_{cj} + E_{crj} + \varepsilon^H_{crj}. \tag{2}$$

A half-specific mean lets the two item sets differ in difficulty, the component E_{crj} is the run deviation shared by the two item populations, and ε^H_{crj} carries the remaining item-dependent variation. For the unbiasedness derivation we assume independent, identically distributed run effects within a configuration, an assumption the released runs only approximate (Section 3), and we write their covariance as $\Sigma_{E,c}$. This target applies to a redrawn or extended item bank. A run's item-specific deviations on the released bank fall into ε, and fixed-bank reproducibility is a separate question.

2.2 Cross-half covariance

We partition each benchmark's items into disjoint halves with a frozen random seed and reuse that assignment for its replicate runs, and we then centre each half score across those runs and write $d^H_{crj} = y^H_{crj} - R^{-1} \sum_s y^H_{csj}$. If the item errors have conditional mean zero and zero covariance between the two halves across the relevant runs and benchmarks, then

$$\mathbb{E}[d^A_{crj} d^B_{crk}] = (1 - 1/R)\Sigma_{E,c}(j, k). \tag{3}$$

Note that Equation 3 covers diagonal entries and doesn't require Gaussian run effects. Disjoint item identifiers alone don't establish the required independence, since shared passages and duplicate content can connect the halves, and we check the one benchmark with documented shared passages, where we keep each passage in one half. Centring removes any item effect that stays constant across runs. A run's deviation on every item of one prompt template recurs on any redrawn bank in that format and belongs to run covariance of the fixed battery. We identify the estimate only up to a deviation tied to items that both halves share.

Cross-format inflation keeps only the diagonal and the covariances between benchmarks of different scoring formats, and it is 0.921 $[0.801, 1.035]$. A uniform component shared across benchmarks that carried the whole margin excess of 0.244 above one would lift it to 1.113, and without cross-format run covariance the upper limit admits at most 0.319 of that excess, although negative cross-format run covariance, as the 0.921 point estimate suggests, could offset a larger component (Appendix C.15). We didn't test a format-following component in DataDecide, and a cross-bank rule tests one tied to the item set in PolyPythias. The fixed-bank inflation reported below needs no halves and no cross-half assumption. Only the split's redrawn-bank target and the per-benchmark run variances rest on it.

We write the average centred score as $g^H_{cr} = K^{-1} \sum_j d^H_{crj}$ and compute

$$T_c = \frac{1}{R-1} \sum_r g^A_{cr} g^B_{cr}, \qquad U_c = \frac{1}{K^2(R-1)} \sum_{j,r} d^A_{crj} d^B_{crj}. \tag{4}$$

The two statistics in Equation 4 have expectations $K^{-2}\mathbf{1}^\top \Sigma_{E,c}\mathbf{1}$ and $K^{-2} \operatorname{tr} \Sigma_{E,c}$. We sum each across configurations, and we divide one sum by the other to obtain

$$\widehat{\Lambda} = \sqrt{\frac{\sum_c T_c}{\sum_c U_c}}, \qquad \Lambda^2 = \frac{\mathbf{1}^\top \Sigma_E \mathbf{1}}{\operatorname{tr} \Sigma_E} = 1 + (K-1)\bar{r}_E, \qquad K_{\text{eff}} = \frac{K}{\Lambda^2}, \tag{5}$$

where $\Sigma_E = N^{-1} \sum_c \Sigma_{E,c}$ is the covariance matrix averaged over the N configurations, whose off-diagonal contribution we express through the effective coefficient

$$\bar{r}_E = \frac{\sum_{j \neq k} \Sigma_E(j, k)}{(K-1) \operatorname{tr} \Sigma_E}. \tag{6}$$

This coefficient equals the mean off-diagonal correlation only when the marginal variances are equal (Appendix A). Here K_{eff} compares this battery with independent benchmarks of the same average variance and doesn't count independent information sources. Since Λ and $\bar{r}_E$ normalise by the battery's own trace, they can rise when the removal of a high-variance benchmark lowers σ_{agg}, and we compare batteries through σ_{agg} (Table 1).

Either pooled sum can be nonpositive in a finite sample, and calibration simulations check the bias of the ratio. A subtraction of item-sampling variance from the full-data diagonal instead needs a model of item-by-run noise, and on the released runs it gives 1.246 $[1.144, 1.340]$ against 1.244 on margins (Appendix D). On the released bank, the replicate standard deviation of the full-bank average gives inflation 1.237 $[1.137, 1.331]$ on margins and 1.067 $[0.994, 1.134]$ on accuracy, against 1.244 and 1.078 from the split. Its denominator sums per-benchmark replicate variances, and any item component the halves share counts as fixed-bank run variance. Under that item-by-run model, item noise is 0.037 of the full-bank margin diagonal and 0.186 of the accuracy one, so the two targets are nearly equal. Without BoolQ, however, it is 0.578 of the accuracy diagonal and fixed-bank accuracy inflation of 1.285 falls below the split's 1.558.

2.3 Interval inference

We compute nominal 95% wild cluster bootstrap-t intervals over the 25 recipe clusters, using 4,999 Rademacher draws and a linearisation of the squared ratio, with the observed denominator held fixed during resampling (Appendix D).

The wild interval's coverage across the twelve populations of Table 8, at 2,000 replicates each, runs from 0.928 to 0.954, and a configuration percentile bootstrap that ignores recipe dependence drops to 0.872 under recipe-shared effects. Within-benchmark cross-half error correlation of 0.1 and the accuracy-like null population bias the estimate down by 0.011 to 0.016. That is, in these populations the downward bias makes an excess harder to find.

3 Data and analysis scope

From the ten DataDecide benchmarks ARC-Challenge and ARC-Easy (Clark et al., 2018), BoolQ (Clark et al., 2019), CommonsenseQA (Talmor et al., 2019), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), Social IQa (Sap et al., 2019), and WinoGrande (Sakaguchi et al., 2020), we analyse 37,682 items per run across 25 recipes at 150M, 300M, 530M, 750M, and 1B parameters, since smaller models lack per-item replicate coverage. Replicate seed labels are $\{2, 14, 15\}$ below 1B and $\{2, 4, 5\}$ at 1B.

In each configuration we select the largest checkpoint step that all three runs share. The release's checkpoint index puts this step at 41.5% of the 750M default runs' final step on average and 87% at 530M. In 26 configurations, one at 530M and 25 at 750M, the largest final step exceeds the smallest by more than a factor of 1.5. At 750M the default runs stop short of their released final models. The auxiliary runs, however, sit near their final steps in all but one configuration.

We interpret the full estimate as covariance among the released runs at the selected steps. The auxiliary-run contrast, at 1.256, removes the batch component. The 26 severely truncated configurations cover the 750M band and one 530M configuration, and without them margin inflation is 1.276 ($[1.114, 1.420]$) and accuracy inflation 1.120 ($[1.020, 1.208]$), a subset we chose after we saw the full-sample estimates. On the 33 configurations that share a final step, margin inflation is 1.082 ($[0.121, 1.530]$) and accuracy 1.129 ($[0.769, 1.397]$), intervals too wide to separate the value one from the full-sample estimates, although the auxiliary contrast on full-sample margins, whose runs sit near their final steps, still excludes one. A configuration's margin influence has Spearman correlation -0.036 with its selected step's share of the default final step (Appendix B.1).

At the adjacent earlier shared step the accuracy interval is $[1.044, 1.210]$ and excludes one, and across the 123 configurations with such a step the paired change in accuracy inflation is 0.048 ($[-0.022, 0.130]$, Appendix C.14). Of course, that check can't see an offset shared by every configuration. The auxiliary contrast also leaves schedule differences in place, a gap the PolyPythias seeds avoid with one shared final step. The accuracy bound thus holds at the scored checkpoint.

Algorithm 1 SNAP measurement recipe for a fixed battery of K benchmarks scored on R replicate runs per configuration.

1: Score every replicate run r of every configuration c on every item of the K benchmarks.
2: Assign each benchmark's items to halves A and B once, with a frozen seed, and reuse that assignment for every run.
3: Average items within each half to obtain y^A_{crj} and y^B_{crj}, then centre across the R runs within each configuration to obtain d^H_{crj}.
4: Form the battery averages g^H_{cr} and the statistics T_c and U_c of Equation 4.
5: Pool over configurations and report $\widehat{\Lambda} = \sqrt{\sum_c T_c / \sum_c U_c}$ and $K_{\text{eff}} = K/\widehat{\Lambda}^2$.
6: Report a wild cluster bootstrap-t interval over the recipe clusters (Appendix D) and repeat the split to measure re-split variation.
7: Recompute $\widehat{\Lambda}$ after removing each benchmark and each cluster in turn, and report that spread beside the interval, since the factor belongs to the battery.

We didn't draw the analysis plan's split of screening and estimation recipes. Consequently, every choice in the primary analysis could see all 125 configurations, and neither the plan's predicted inflation thresholds nor its gain-simulator check G5 held (Appendix B).

The primary estimand keeps all ten benchmarks, even those whose estimated diagonal covariance is negative, since an unbiased moment estimate can be negative near zero and the deletion of such a benchmark would change the estimand.

We add a second model family whose replicate design DataDecide lacks. PolyPythias (van der Wal et al., 2025) trains nine seeds at each of 14M, 31M, 70M, 160M, and 410M parameters with shared code, data, and hyperparameters and a seed that sets initialisation and data order together, and we score all 45 runs at the shared final step 143,000 on all 37,682 items. We took the request text from the DataDecide release, and every prompt matches its scoring byte for byte. Appendix C.18 reports an earlier transfer check on 4,755 items.

For the identification check we built a second bank of 6,808 items from the same ten benchmarks with OLMES at a pinned commit, with the train split for eight benchmarks and the validation split for OpenBookQA and MMLU. We dropped any item whose context matches a bank-one context or whose question appears among its own few-shot exemplars, and we capped each benchmark apart from MMLU at 600 items. We committed both PolyPythias rules before we scored either bank.

4 Results

4.1 Full-battery estimates and composition

Full-battery margins exclude independence, and accuracy gives only a bound. At the accuracy upper limit of 1.157 the battery has $\bar{r}_E$ of 0.038 and K_{eff} of 7.47 under the model.

Table 1. Inflation estimates across 125 configurations with nominal 95% recipe-cluster intervals, or cluster-robust $t(4)$ intervals over the five size bands where marked. The auxiliary contrast compares the two auxiliary runs and removes the default-versus-auxiliary batch component. The aggregate standard deviation σ_{agg} of the battery mean under the fitted covariance (Appendix A) uses margin units for margins and fractions for accuracy.

Score	Estimate	$\widehat{\Lambda}$	Interval	K_{eff}	σ_{agg}
Margin	Raw	1.244	[1.143, 1.338]	6.46	0.00572
Margin	Auxiliary contrast	1.256	[1.090, 1.400]	6.34	0.00562
Margin	Size-band clusters	1.244	[1.030, 1.426]	6.46	0.00572
Accuracy	Raw	1.078	[0.993, 1.157]	8.60	0.00514
Accuracy	Auxiliary contrast	1.084	[0.939, 1.207]	8.51	0.00516
Accuracy	Size-band clusters	1.078	[0.961, 1.184]	8.60	0.00514

Table 5 and Appendix C give the per-benchmark diagnostics, which don't enter the headline ratio because it uses covariance products directly.

The factor belongs to this battery, since margin correlations track scoring format (cloze for HellaSwag, PIQA and WinoGrande, yes/no for BoolQ, and multiple choice for the other six), with means of 0.624 within formats and 0.007 across them when we exclude WinoGrande's undefined correlations (Figure 3 in Appendix C). In particular, BoolQ, the only benchmark in its format, carries 68% of the margin covariance trace and 84% of the accuracy trace, and its removal raises margin inflation to 1.786 $[1.689, 1.877]$, and accuracy inflation to 1.558 $[1.412, 1.687]$. The removal of CommonsenseQA instead gives inflation estimates of 1.096 and 1.014. The removal estimates imply off-diagonal terms with BoolQ that sum to about -0.15 of the margin trace and -0.06 of the accuracy trace. Its removal lowers the denominator of the ratio more than the numerator and so raises the factor even as the margin aggregate standard deviation falls from 0.00572 to 0.00516. A BoolQ weight of 0.0867 in place of a tenth already gives an accuracy interval that excludes one (Appendix C).

Shared passages don't account for BoolQ's trace share, but that check can't reach other shared item effects. We recovered the 2,938 passages behind BoolQ's 3,270 questions from the release's request files, and when we assign each passage whole to one half, the two estimates change by -0.0002 and -0.0001 on average across fifty re-splits. The BoolQ diagonal moves by 0.03% and 0.15% under the same reassignment. A permutation that reshuffles each run's item deviations inside each of the 66 item groups, MMLU's 57 subjects and the other nine benchmarks, leaves all trait scores fixed and breaks the tie between particular items and runs. Across 2,000 such permutations the permuted estimates sit about 0.001 below the observed margin value and 0.002 below accuracy. Item-level sharing thus seems to account for little of either, but an effect carried by every item of a benchmark survives it.

Benchmark removal moves the estimates more than recipe or size removal. Recipe removal moves margin inflation between 1.207 and 1.256, and all 300 recipe-pair deletions keep the interval above one, with estimates from 1.177 to 1.263 and no lower limit below 1.037 (Appendix C). Size removal moves it between 1.178 and 1.289, where the deletion of 530M gives $[0.993, 1.338]$, an interval that includes one. Figure 2 shows every single-benchmark removal with its interval on both scales, and Appendix C.10 separates benchmark count, identity, and weights. We should stress that the design treats this battery as fixed, and that spread isn't a distribution over benchmark collections.

4.2 A PRE-SPECIFIED TEST ON HELD-OUT TASKS

The pre-specified rule fails on the four held-out tasks. Held-out margin inflation on the 225 runs is 1.217 $[0.872, 1.498]$ against 1.240 $[1.054, 1.397]$ for the original ten, a comparison specified before any held-out score existed. Recall that inflation belongs to its battery, which is why the four-task and ten-task values aren't directly comparable. The held-out interval is also 0.626 wide against 0.343 for the original, and the closer like-for-like reference is the four-benchmark subset row of Table 2, with a median of 1.176, although it pools all five sizes where the held-out test uses three.

The protocol named SciQ, MedMCQA, and two AGIEval (Zhong et al., 2024) reasoning tasks, with a pre-specified rule that replaced any task whose 1B pilot accuracy fell below chance plus 0.02, a bar both AGIEval tasks narrowly missed (Appendix C.22). The battery became SciQ, MedMCQA, and multiple-choice versions of DROP and CoQA, with 1,000 items in each of the four tasks and items sharing a passage or story kept in the same half. Our GPU budget couldn't cover all 375 released checkpoints, and our protocol's scope rule limited the test to those runs at the steps of the primary analysis. The test passes only if the lower 95% limit exceeds one, and Table 2 reports the test at its pre-specified scope. Every partial scope read before the full one also failed the rule, with held-out margin estimates of 1.152 to 1.203, and Appendix C.22 lists all five, with simulated power by number of sizes.

As an exploratory result, the held-out accuracy interval excludes one while the margin interval doesn't, the reverse of the original battery's pattern on the same runs. Accuracy inflation is 1.220 $[1.066, 1.359]$ against 1.097 for the original ten on the same runs. Without CoQA, held-out margin inflation falls to 1.090, and without any other task it stays between 1.120 and 1.213. Only the removal of DROP, at 1.213 $[1.035, 1.377]$, excludes one. The held-out estimate thus appears to depend on battery composition as the original does.

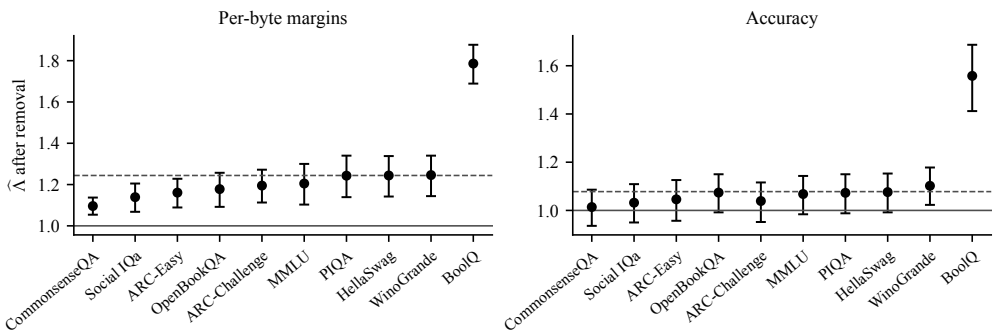


Figure 2. Inflation after removing each benchmark in turn, with nominal 95% recipe-cluster intervals. The solid line marks one and the dashed line the full ten-benchmark estimate. On margins every removal leaves an interval that excludes one, with estimates from 1.096 without CommonsenseQA to 1.786 without BoolQ.

Table 2. The pre-specified test at 530M, 750M, and 1B, with nominal 95% recipe-cluster intervals over 75 configurations. The auxiliary contrast compares the two auxiliary runs, the original rows read the released runs at the same sizes as a comparison specified before any held-out score existed, and the last row gives the median over 60 four-benchmark subsets of the original battery across all five sizes with subset 5th to 95th percentiles, not an interval, from the committed battery-size curve.

Battery	Score	$\widehat{\Lambda}$	Interval
Held-out, four tasks	Margin	1.217	[0.872, 1.498]
Held-out, four tasks	Accuracy	1.220	[1.066, 1.359]
Held-out, auxiliary contrast	Margin	1.181	[0.906, 1.410]
Original, ten benchmarks	Margin	1.240	[1.054, 1.397]
Original, ten benchmarks	Accuracy	1.097	[0.922, 1.246]
Original, auxiliary contrast	Margin	1.273	[0.738, 1.629]
Original, four-benchmark subsets, all sizes (percentiles)	Margin	1.176	0.982 to 1.476

4.3 A SECOND FAMILY AND A DISJOINT ITEM BANK

Two pre-specified rules decide what the 45 runs show. The first reads margin inflation across the 45 PolyPythias runs from a delete-one-seed jackknife with eight degrees of freedom and passes when the lower endpoint exceeds one, since five size clusters are too few for the wild bootstrap. The second puts every bank-one item in half A and every bank-two item in half B on the same runs, and that assignment removes any component the halves share only through a common item set. It returns not supported when that cross-bank interval sits above one and the jackknife interval for the within-minus-cross difference in log squared inflation includes zero, supported when that difference interval lies above zero and the cross-bank interval covers one, and undecided for any other pattern. Any covariance from the banks' other differences also enters the cross-bank estimate, since they differ in item count and exemplar overlap and bank two draws most of its items from train splits, which pretraining data are more likely to contain.

Over 4,000 replicates that we simulated after we fixed both rules, with seed effects and item noise drawn from the DataDecide margin fit (Appendix C.18), the first rule passes in 0.259 at a true 1.244 and in 0.016 at independence, with the middle 90% of estimates between 1.026 and 1.456. At a true 1.244 with no shared item component the second returns not supported in 0.226, supported in 0.013 and undecided in 0.761, while under independence a uniform item share of 0.124, which reproduces 1.244 only in the population of Appendix C.15 and lifts the median within-bank estimate to 1.046, gives supported in 0.033 and undecided in 0.957. At a share of 0.755, enough alone to give 1.244 on this noise, the second returns supported in 0.067, not supported in 0.015 and undecided in 0.919. An undecided verdict from the second rule can't exclude one.

The second family doesn't pass the margin rule's threshold. Margin inflation across the 45 runs is 1.302 with a jackknife interval $[0.921,\ 1.595]$ that includes one, and accuracy gives 1.481 with interval $[0.561,\ 2.019]$. Five sizes with nine seeds each give 40 within-configuration contrasts against 250 in DataDecide, too few for the interval to separate this family's inflation from one or from DataDecide's 1.244. The per-size estimates run from 0.997 at 160M to 1.576 at 410M (Appendix C.18).

The identification check returns undecided under its rule. With bank one as half A and bank two as half B on the same 45 runs, margin inflation is 1.293 with interval $[0.920,\ 1.580]$, and the within-minus-cross difference in log squared inflation has interval $[-0.035,\ 0.065]$. Bank two holds under a fifth of bank one's items, and that smaller bank widens both intervals.

4.4 Concentration by scoring format

Format and task content aren't separable in this battery. If we set within-format correlations to their mean of 0.624 and cross-format ones to 0.007 and keep the benchmark seed variances, margin inflation is 1.285 against the observed 1.244, and 1.728 against 1.786 without BoolQ. The fit is in sample, however, and Appendix C.5 checks it with each benchmark held out.

4.5 Predicting aggregate uncertainty

Covariance models beat independence on margins only when given held-out marginal seed standard deviations. We compared five covariance models and an exploratory rescaled plug-in of observed score correlations on five recipe folds, with the run-covariance models fitted off fold (Table 7 in Appendix C). On margins the rank-one, rank-one plus gain, and full models lower mean squared error of log aggregate standard deviation by about 0.03 from the 0.0358 of independence, and their paired recipe-bootstrap intervals exclude zero. The rank-one interval, however, reaches zero when we fill nonpositive held-out variances from training folds. On accuracy every such interval includes zero, and an operational comparison that refits without held-out marginals separates no model from independence on either scale.

On real comparisons at each size below 1B, we called a pair of recipes when their aggregate gap exceeded two standard errors, computed either under independence or with the estimated covariance, and neither estimate uses the pair. We scored each call against the pair's ordering at 1B whenever that ordering itself cleared two corrected standard errors. Across the four sizes and both scales, the correction removes between 1 and 12 of the 173 to 223 calls on such pairs and changes the share of wrong calls by at most 0.009, from rates between 0.029 and 0.070, and every recipe-bootstrap interval for that change reaches zero (Table 12 in Appendix C.23). Note that a wrong call also counts genuine rank changes between sizes, and these rates aren't false-positive rates in the usual sense.

On PolyPythias null seed pairs, with deviations from the other seven seeds of one configuration, both standard errors, under independence and with the estimated covariance, call 0.133 of margin pairs at two standard errors against a nominal 0.046, or 0.092 for t with six degrees of freedom (Appendix C.24). In single-run comparisons the correction changes the wrong-call rate by at most 0.019 for runs sharing a seed label and 0.016 for runs that don't, and every interval reaches zero. The ratio of paired-difference standard deviations for seed-matched and seed-unmatched pairs runs from 0.928 to 1.025 and gives no sign that shared seed labels change the correction's effect.

5 Related work

Bouthillier et al. (2021) derive variance inflation from correlated trainings on one task, whereas we account for covariance across benchmarks within a replicate run. Reimers and Gurevych (2017), Zhou et al. (2020), Summers and Dinneen (2021), Madaan et al. (2024), and Fehlauer et al. (2025) study seed-related variability, Sellam et al. (2022) release 25 pretraining seeds of one BERT configuration, and Dodge et al. (2020) test whether good initialisations transfer across tasks. Jordan (2024) finds run deviations in accuracy nearly independent across shifted ImageNet test sets, a pattern our accuracy interval allows. Our margin covariance, by contrast, sits within scoring formats. Miller (2024) gives clustered and paired item-sampling errors with each model treated as fixed, and Henderson et al. (2018) and Agarwal et al. (2021) resample runs within a task and discard the paired run

effects we keep. Apart from such shifted test sets, we know of no study that estimates the covariance of run noise between distinct benchmarks or an effective benchmark count from that covariance.

Our cross-half estimator adapts the split-half construction of Spearman (1910); Brown (1910), the attenuation correction of Spearman (1904), and the variance components of Searle et al. (1992) and generalizability theory (Brennan, 2001), and we add the estimand, cross-benchmark run covariance at a fixed configuration, which neither Miller (2024) nor Heineman et al. (2025) estimates, and its measurement on released runs. Cheverud (1988) motivates our phenotypic plug-in (Table 7), and we take the covariance analogy of Martin and Eaves (1977) without its variance components.

The few-cluster inference follows Cameron et al. (2008) and MacKinnon and Webb (2017). Heineman et al. (2025) relate per-benchmark signal and noise on DataDecide and OLMo to decision accuracy, and our ratio computed from their checkpoint noise gives DataDecide's accuracy average 1.104, near our 1.078, and OLMo 2's 0.992 against a median seed value of 0.986 over its last 30 checkpoints (Appendix C.20). Across models, benchmark scores share low-dimensional structure (Ruan et al., 2024; Burnell et al., 2023b), whereas we measure covariance of run deviations within a configuration.

6 Discussion and limitations

We read the accuracy result, which leaves independence open, as an upper bound of 1.157 on inflation at the scored checkpoint, and at accuracy-like item noise this design detects a true 1.10 in 0.135 of replicates. Exploratory cuts that exclude one, without BoolQ at 1.558, without the 26 truncated configurations at 1.120, at the adjacent checkpoint and on the held-out tasks, don't settle it.

The study is exploratory apart from the held-out test and the two PolyPythias rules, which we fixed before scoring, and the held-out test and the first rule both failed. The shared-item check is unrun in the family that supplies our headline, whose 1.244 thus stays untested on fresh items. We score the default 750M runs at 41.5% of their final step on average. The 33 configurations that share a final step give a margin interval too wide to test the full estimate.

The populations stop at 1B and at likelihood-scored multiple choice, since no larger public multi-seed population shares replicate item scores (Appendix B) and generative tasks lack our margin. On accuracy a uniform item effect shared across benchmarks could reproduce 1.078 with a cross-format value of 1.047 inside $[0.848,\ 1.063]$, and a format-following component remains unmeasured in DataDecide (Appendix C.15). The cross-bank check in PolyPythias leaves that item effect undecided in one family.

We cluster intervals on recipes, but the release shares one seed label across every recipe inside a size band, and a run effect following the band would take simulated margin coverage to 0.821 at a quarter of latent variance and 0.580 at a half, although the size-band-clustered margin interval $[1.030,\ 1.426]$ of Table 1 still excludes one. A permutation test that aligns default and auxiliary runs across recipes gives that effect an upper bound of 0.044 of margin run covariance that can understate it, where simulated coverage is 0.941 at a share of 0.05, and its power is 0.490 at the bound.

Against one configuration's six held-out seeds, a noisy target, independence predicts margin inflation better than either three-seed fit, and SNAP beats it only against a target pooled across configurations, which resembles Section 4.5's recipe folds, and not on five-size margins (Appendix C.18).

For a fixed released bank the replicate standard deviation of the battery average gives the aggregate uncertainty directly, and we recommend that authors report it with a benchmark-removal check. The split adds the item-general value for a redrawn or extended bank, the per-benchmark run variances, and K_{eff} for planning run counts. Without replicates, the format fit of Appendix C.5 gives a fallback untested outside this battery. In an equicorrelated population at the observed $\bar{r}_E$ of 0.061 on margins and 0.018 on accuracy, with generic item noise, three runs each let the margin interval exclude one in 0.997 of replicates and accuracy needs eight to reach 0.803, but at the accuracy battery's own noise three runs detect a true 1.078 in 0.111 of replicates (Appendix C). The next step is to score the 375 DataDecide runs on the second bank.

AI USE STATEMENT

We used generative AI tools for three purposes. They helped polish the wording of the manuscript, they helped us find related work, and they gave minor help with analysis code. The authors designed the study, wrote the paper, checked every AI suggestion before using it, and take full responsibility for the code, the citations and the final text.

ETHICS STATEMENT

This study analyses released model outputs and introduces no new participant recruitment or data collection. Its intended use is to improve uncertainty reporting in model evaluation. An independence approximation used outside the studied battery could understate uncertainty, which keeps the recommendations conditional on the measured population and score definition. Use of derived artifacts must retain the applicable source attribution and licensing requirements.

REPRODUCIBILITY STATEMENT

Section 2 and Appendix A define the estimator and its assumptions. Appendix D specifies the interval calculation and data reduction. Appendix B records the populations and planned checks. Appendix C reports later sensitivity analyses alongside the primary results. The released archive also holds full-length versions of the compressed appendix sections. The primary calculation uses 4,999 bootstrap draws, 2,000 calibration replicates, 2,000 replicates for each of the item-group and batch-slot permutation checks and 500 for the run-label permutation bias check, 50 item re-splits, and master seed 20260101, and reproduction requires the per-item reductions, item ordering and half assignments, selected checkpoint identifiers, and the corresponding analysis scripts. We release all four of them alongside the paper. Neither the analysis plan nor the protocol and rule commits carry an independently corroborated timestamp. A separate implementation run on Kaggle reproduces the primary estimates, the BoolQ-removal estimates, and their intervals to three decimals from the released per-item scores, a run that also produced the passage-aware split, the coverage simulation, the operational prediction comparison, and the seed-matched pair analysis. The earlier 4,755-item PolyPythias transfer check, its five-size extension, and the checkpoint results come from execution records that we didn't rerun for the final analysis. The pre-specified test ships with its protocol, the dated amendments that record each change to it, the frozen hash of its request file, and the kernels that scored and analysed it, and the last twenty runs at 1B were scored off Kaggle on three RTX 3090 cards. We score the PolyPythias battery on both banks in float32 with transformers 4.57.1 on Kaggle T4 cards, each run checking its cached scores against an uncached reference on 24 items to within 0.001 nats, and the release will include the hashes of both request files, the bank-two file as a deterministic rebuild from its manifest, with the scoring kernels. Separate CPU kernels on the released runs produced the recipe-decision, seed-count, battery-size, and item-count analyses. Fifteen further CPU kernels behind later sensitivity analyses, among them the null seed-pair, split-against-replicate, rule-power and weighting checks, ship with their scripts, logs and outputs in the archive's rebuttal_checks folder.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In Advances in Neural Information Processing Systems, volume 34, pp. 29304–29320, 2021.

Robert M. Bell and Daniel F. McCaffrey. Bias reduction in standard errors for linear regression with multi-stage samples. Survey Methodology, 28(2):169–181, 2002.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In International Conference on Machine Learning, volume 202, pp. 2397–2430, 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05):7432–7439, 2020.

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

Robert L. Brennan. Generalizability Theory. Statistics for Social Science and Public Policy. Springer, New York, 2001.

William Brown. Some experimental results in the correlation of mental abilities. British Journal of Psychology, 3(3):296–322, 1910.

Nghia Bui, Guergana Savova, and Lijing Wang. Assessing the macro and micro effects of random seeds on fine-tuning large language models. In Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 41–46, 2025. arXiv:2503.07329.

Ryan Burnell, Wout Schellaert, John Burden, Tomer D. Ullman, Fernando Martinez-Plumed, Joshua B. Tenenbaum, Danaja Rutar, Lucy G. Cheke, Jascha Sohl-Dickstein, Melanie Mitchell, Douwe Kiela, Murray Shanahan, Ellen M. Voorhees, Anthony G. Cohn, Joel Z. Leibo, and Jose Hernandez-Orallo. Rethink reporting of evaluation results in AI. Science, 380(6641):136–138, 2023a.

Ryan Burnell, Han Hao, Andrew R. A. Conway, and Jose Hernandez-Orallo. Revealing the structure of language model capabilities. arXiv:2306.10062, 2023b.

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

Donald T. Campbell and Donald W. Fiske. Convergent and discriminant validation by the multitrait-multimethod matrix. Psychological Bulletin, 56(2):81–105, 1959.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. With little power comes great responsibility. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 9263–9274, 2020.

James M. Cheverud. A comparison of genetic and phenotypic correlations. Evolution, 42(5):958–968, 1988.

James M. Cheverud. A simple correction for multiple comparisons in interval mapping genome scans. Heredity, 87(1):52–58, 2001.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. arXiv:1803.05457, 2018.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 2924–2936, 2019.

Lee J. Cronbach. Coefficient alpha and the internal structure of tests. Psychometrika, 16(3):297–334, 1951.

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

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

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. arXiv:2002.06305, 2020.

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. Replicability analysis for natural language processing: Testing significance with multiple datasets. Transactions of the Association for Computational Linguistics, 5:471–486, 2017.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 2368–2378, 2019.

Nicolas Dufour, Alexei A. Efros, and Patrick Pérez. The FID lottery: Quantifying hidden randomness in generative-model evaluation. arXiv:2606.20536, 2026.

Finlay Fehlauer, Kyle Mahowald, and Tiago Pimentel. Convergence and divergence of language models under different random seeds. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pp. 32970–32979, 2025. arXiv:2509.26643.

Kyle Gorman and Steven Bedrick. We need to talk about standard splits. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2786–2791, 2019.

Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. OLMES: A standard for language model evaluations. In Findings of the Association for Computational Linguistics: NAACL 2025, pp. 5020–5048, 2025.

Michael Hagmann, Philipp Meier, and Stefan Riezler. Towards inferential reproducibility of machine learning research. In International Conference on Learning Representations, 2023.

Michael Hardy, Anka Reuel, Lijin Zhang, Jodi M. Casabianca, Sang Truong, Yash Dave, Hansol Lee, Benjamin Domingue, and Sanmi Koyejo. AI cartography: Mapping the latent landscape of AI benchmark ecosystems. arXiv:2605.25272, 2026.

Andreas Haupt, Justin Hartenstein, Anka Reuel, Mykel Kochenderfer, and Sanmi Koyejo. Welfare, improvability, and variance: A principal-agent approach to optimal benchmark item aggregation. arXiv:2605.30916, 2026.

Larry V. Hedges, Elizabeth Tipton, and Matthew C. Johnson. Robust variance estimation in meta-regression with dependent effect size estimates. Research Synthesis Methods, 1(1):39–65, 2010.

David Heineman, Valentin Hofmann, Ian Magnusson, Yuling Gu, Noah A. Smith, Hannaneh Hajishirzi, Kyle Lo, and Jesse Dodge. Signal and noise: A framework for reducing uncertainty in language model evaluation. In Advances in Neural Information Processing Systems, 2025. arXiv:2508.13144.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1):3207–3214, 2018.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In International Conference on Learning Representations, 2021.

Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. In Conference on Language Modeling (COLM), 2025.

Valentin Hofmann, David Heineman, Ian Magnusson, Kyle Lo, Jesse Dodge, Maarten Sap, Pang Wei Koh, Chun Wang, Hannaneh Hajishirzi, and Noah A. Smith. Fluid language model benchmarking. In Conference on Language Modeling, 2025. arXiv:2509.11106.

Keller Jordan. On the variance of neural network training with respect to test sets and distributions. In International Conference on Learning Representations, 2024.

Siddharth Karamcheti, Laurel Orr, Jason Bolton, Tianyi Zhang, Karan Goel, Avanika Narayan, Rishi Bommasani, Deepak Narayanan, Tatsunori Hashimoto, Dan Jurafsky, Christopher D. Manning, Christopher Potts, Christopher Ré, and Percy Liang. Mistral: A journey towards reproducible language model training. Stanford Center for Research on Foundation Models, 2021. `https://crfm.stanford.edu/2021/08/26/mistral.html`.

Alex Kipnis, Konstantinos Voudouris, Luca M. Schulze Buschoff, and Eric Schulz. metabench, a sparse benchmark of reasoning and knowledge in large language models. In International Conference on Learning Representations, 2025. arXiv:2407.12844.

Leslie Kish. Survey Sampling. John Wiley & Sons, New York, 1965.

Guneet Kohli. Nine judges, two effective votes: Correlated errors undermine LLM evaluation panels. arXiv:2605.29800, 2026.

J. Li and L. Ji. Adjusting multiple testing in multilocus analyses using the eigenvalues of a correlation matrix. Heredity, 95(3):221–227, 2005.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, pp. 3622–3628, 2020.

James G. MacKinnon and Matthew D. Webb. Wild bootstrap inference for wildly different cluster sizes. Journal of Applied Econometrics, 32(2):233–254, 2017.

Lovish Madaan, Aaditya K. Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. Quantifying variance in evaluation benchmarks. arXiv:2406.10229, 2024.

Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D. Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, Noah A. Smith, Pang Wei Koh, and Jesse Dodge. DataDecide: How to predict best pretraining data with small experiments. In International Conference on Machine Learning, 2025. arXiv:2504.11393.

N. G. Martin and L. J. Eaves. The genetical analysis of covariance structure. Heredity, 38(1):79–95, 1977.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pp. 217–227, 2020.

Solomon Messing. Hidden measurement error in LLM pipelines distorts annotation, evaluation, and benchmarking. arXiv:2604.11581, 2026.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2381–2391, 2018.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. arXiv:2411.00640, 2024.

Dale R. Nyholt. A simple correction for multiple testing for single-nucleotide polymorphisms in linkage disequilibrium with each other. American Journal of Human Genetics, 74(4):765–769, 2004.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Conference on Health, Inference, and Learning, volume 174 of Proceedings of Machine Learning Research, pp. 248–260, 2022.

Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. Transactions of the Association for Computational Linguistics, 7:249–266, 2019.

Nils Reimers and Iryna Gurevych. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 338–348, 2017.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of language model performance. In Advances in Neural Information Processing Systems, volume 37, 2024. arXiv:2405.10938.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05):8732–8740, 2020.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 4463–4473, 2019.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In Advances in Neural Information Processing Systems, volume 36, 2023.

Shayle R. Searle, George Casella, and Charles E. McCulloch. Variance Components. John Wiley & Sons, New York, 1992.

Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, and Ellie Pavlick. The MultiBERTs: BERT reproductions for robustness analysis. In International Conference on Learning Representations, 2022.

Tommy Sha and Stella Zhao. BenchScope: How many independent signals does your benchmark provide? arXiv:2603.29357, 2026.

C. Spearman. The proof and measurement of association between two things. The American Journal of Psychology, 15(1):72–101, 1904.

C. Spearman. Correlation calculated from faulty data. British Journal of Psychology, 3(3):271–295, 1910.

Cecilia Summers and Michael J. Dinneen. Nondeterminism and instability in neural network optimization. In Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pp. 9913–9922, 2021.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4149–4158, 2019.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, et al. 2 OLMo 2 furious. arXiv:2501.00656, 2025a. Shorter version in Conference on Language Modeling (COLM) 2025.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, et al. Olmo 3. arXiv:2512.13961, 2025b.

Oskar van der Wal, Pietro Lesci, Max Müller-Eberstein, Naomi Saphra, Hailey Schoelkopf, Willem Zuidema, and Stella Biderman. PolyPythias: Stability and outliers across fifty language model pre-training runs. In International Conference on Learning Representations, 2025. arXiv:2503.09543.

Siyuan Wang, Zhongkun Liu, Wanjun Zhong, Ming Zhou, Zhongyu Wei, Zhumin Chen, and Nan Duan. From LSAT: The progress and challenges of complex reasoning. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30:2201–2216, 2022.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Proceedings of the 3rd Workshop on Noisy User-generated Text, pp. 94–106, 2017.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4791–4800, 2019.

Yang Zhao, Jiahao Lu, Bin Huang, Guhua Zhang, and Jie Zhou. Most transformer modifications still do not transfer at 1–3B: A 2020–2026 update to Narang et al. (2021) with downstream evaluation and a noise floor. arXiv:2605.20798, 2026.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A human-centric benchmark for evaluating foundation models. In Findings of the Association for Computational Linguistics: NAACL 2024, pp. 2299–2314, 2024.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. The curse of performance instability in analysis datasets: consequences, source, and suggestions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8215–8228, 2020.

A Derivations and interpretation

A.1 What SNAP estimates

SNAP abbreviates Seed Noise Across Phenotypes, and a phenotype here denotes a benchmark score that the rest of the paper calls a trait. Our estimation target is the covariance of run deviations that generalise across item halves. DataDecide replicates vary initialisation and data order together, and auxiliary runs below 1B use shortened training schedules. Common-step scoring aligns the observation step, while schedule differences remain within the full-set target. We therefore use run covariance for our empirical estimate, and we keep a seed-only interpretation for designs with matched training conditions.

The item-general estimate can differ from the run variance of a fixed evaluation bank. A run can perform differently on particular items and keep its expected score over an item population, and under the measurement model that variation enters the item error and doesn't contribute to the cross-half covariance target. Repeated evaluation of a fixed item bank can retain this component. Its run variance needn't equal the item-general variance estimated here.

The analogy with quantitative genetics distinguishes covariance between configurations from residual covariance within them, and it doesn't identify separate additive-genetic and shared-environment components in a model-training experiment. Between-configuration correlation R_P reflects differences in recipe and model size, and within-configuration covariance Σ_E concerns replicate deviations. The plug-in comparison asks whether the first correlation structure predicts the second once we supply marginal run variances.

A.2 Notation

Table 3. Notation for the cross-half construction and its empirical diagnostics. The coefficient $\bar{r}_E$ uses covariance normalisation, which differs from mean correlation under unequal marginal variances.

Symbol	Meaning
c, r, j, i	Configuration, replicate run, benchmark, and item indices
N, R, K	Configuration, replicate, and benchmark counts, with values 125, 3, and 10
$\Sigma_{E,c}, \Sigma_E$	Configuration covariance and its average across the design
R_E, R_P	Run-effect and between-configuration score correlation matrices
$y^{\mathrm{marg}}, y^{\mathrm{acc}}$	Mean per-byte margin and accuracy on the specified item set
d^H_{crj}, g^H_{cr}	Centred benchmark score and its benchmark average on half H
T_c, U_c	Cross-half aggregate covariance and its diagonal-only contribution
$\Lambda, \bar{r}_E$	Standard-deviation inflation and effective covariance coefficient
Λ_M, Λ_A	The same inflation computed on margins and on accuracy
K_{eff}	Independent-benchmark equivalent at the same marginal RMS deviation
$\sigma_{\mathrm{agg}}, \sigma_{\mathrm{indep}}$	Aggregate deviation with full covariance and under independence
$\rho_g, \bar{v}$	Aggregate reliability and its scaled item-noise contribution
$s_{cr}, x^{(-j)}_{cr}$	Gain covariate and opposite-half competence proxy

A.3 Cross-half identity and its assumptions

Fix a configuration and suppress its index for clarity. Write the half-score model as $y^H_{rj} = \mu^H_j + E_{rj} + \varepsilon^H_{rj}$, so centring across runs gives

$$d^H_{rj} = (E_{rj} - \bar{E}_{\cdot j}) + (\varepsilon^H_{rj} - \bar{\varepsilon}^H_{\cdot j}). \tag{7}$$

Assume that run-effect vectors are independent draws with covariance Σ_E. Conditional on those effects, assume zero-mean item errors and zero covariance between ε^A_{rj} and ε^B_{sk} for the run and benchmark pairs in the calculation. The expansion of $d^A_{rj} d^B_{rk}$ then removes its error cross terms in

expectation and leaves

$$\mathbb{E}[d^A_{rj} d^B_{rk}] = \mathbb{E}[(E_{rj} - \bar{E}_{\cdot j})(E_{rk} - \bar{E}_{\cdot k})] \tag{8}$$

$$= (1 - 1/R)\Sigma_E(j,k). \tag{9}$$

A sum across runs divided by $R-1$ gives the unbiased covariance moment. We symmetrise the two half orderings for the matrix estimate,

$$\widehat{\Sigma}_E(j,k) = \frac{1}{2N(R-1)} \sum_{c,r} \left(d^A_{crj} d^B_{crk} + d^A_{crk} d^B_{crj} \right). \tag{10}$$

Symmetry follows from this construction, while positive semidefiniteness doesn't follow for a finite sample. A negative diagonal moment near zero doesn't by itself falsify the identity, although it prevents ordinary correlation normalisation for that trait.

Different item identifiers can still refer to items with dependent errors, because shared source passages, duplicate questions, and overlapping content can create cross-half covariance. A fixed finite item bank split without replacement also requires a model for its sampling dependence. The available item-half assignment alone can't verify these conditions.

BoolQ has documented shared passages, and we check them directly through the release's request files that carry each question's passage. All 3,270 question passages match the google/boolq validation split exactly, and they give 2,938 passages with 578 questions in passages of two or more. A passage-aware split assigns whole passages to halves and then averages questions. Half sizes therefore differ by a few items, and the equal-half derivation doesn't hold exactly. On one such split the margin and accuracy estimates are 1.248 and 1.097 against 1.244 and 1.078 on the frozen split. Across fifty paired re-splits the passage-aware and item-level rules differ by -0.0002 and -0.0001 on average, with standard deviations 0.0017 and 0.0060.

The BoolQ diagonal moves by 0.03 percent and 0.15 percent. A simulation matched to the observed passage-size histogram adds a run-by-passage interaction shared by the questions on a passage, using 500 replicates at each of five interaction scales, with standard deviation from zero up to the item-noise scale. Across those scales the passage-level and item-level splits differ in Λ by between -0.0008 and 0.0022 on average, within Monte Carlo error of zero. The observed histogram puts about 240 same-passage question pairs across the halves out of about 2.7 million cross-half pairs, and a leaked interaction can't move the diagonal by more than about 0.01 percent even at that scale. The null result above is what this benchmark's passage structure implies, and it isn't evidence of a sensitive test. Outside BoolQ, the request files show two question stems shared across traits and repeated question stems within benchmarks, for example 62 of 1,838 PIQA items, which we haven't grouped.

For a diagonal additional covariance D with nonnegative trace, the inflation ratio becomes

$$\Lambda^2_D = \frac{\mathbf{1}^\top \Sigma_E \mathbf{1} + \operatorname{tr} D}{\operatorname{tr} \Sigma_E + \operatorname{tr} D}. \tag{11}$$

This ratio lies between Λ^2 and one when the denominators are positive, and that explains the attenuation toward one under that restricted model. However, no general upper or lower bound follows once shared-item effects also contribute off-diagonal covariance. Bounding fixed-battery inflation by the item-general factor needs the same restriction.

A.4 Covariance coefficient and effective count

Let $\sigma_j^2 = \Sigma_E(j,j)$ and $r_{jk} = \Sigma_E(j,k)/(\sigma_j \sigma_k)$ whenever both variances are positive. For this finite battery, the variance ratio therefore satisfies

$$\Lambda^2 = 1 + \frac{\sum_{j \neq k} r_{jk} \sigma_j \sigma_k}{\sum_j \sigma_j^2} = 1 + (K-1)\bar{r}_E. \tag{12}$$

The normalised weighted mean correlation instead divides the numerator by $\sum_{j \neq k} \sigma_j \sigma_k$, and for a positive denominator the two coefficients are related by

$$\bar{r}_E = a_\sigma \bar{r}_{\text{weighted}}, \qquad a_\sigma = \frac{\sum_{j \neq k} \sigma_j \sigma_k}{(K-1) \sum_j \sigma_j^2}. \tag{13}$$

For nonnegative standard deviations, a_σ lies between zero and one, and both coefficients coincide once the marginal variances are equal. Two perfectly correlated traits with standard deviations one and two give an effective coefficient of 0.8 against a weighted mean correlation of one.

With average marginal variance $\bar{\sigma}_E^2 = K^{-1} \operatorname{tr} \Sigma_E$, the variance of the benchmark average satisfies

$$\sigma_{\mathrm{agg}}^2 = \bar{\sigma}_E^2 \left[\frac{1}{K} + \frac{K-1}{K} \bar{r}_E \right]. \tag{14}$$

Independent benchmark deviations at the same average variance give $\sigma_{\mathrm{indep}}^2 = \bar{\sigma}_E^2 / K$, making the factor we report $\Lambda = \sigma_{\mathrm{agg}} / \sigma_{\mathrm{indep}}$. A nonzero asymptotic floor follows only for a sequence of batteries whose average variance and effective coefficient converge to suitable limits, and bounded variances alone don't imply that convergence. Our present ten-benchmark study doesn't estimate such a sequence, and we use the finite-battery identity for the reported effective counts.

Our reported effective margin coefficients are 0.0608, 0.0695, and 0.0641 for the raw, gain-adjusted, and auxiliary-contrast rows, with corresponding accuracy coefficients of 0.0181, 0.0225, and 0.0194, retaining the covariance normalisation of Equation 6.

For positive aggregate variance, $K_{\mathrm{eff}} = K / \Lambda^2$ compares the measured average with an independent average having the same marginal RMS deviation. Negative aggregate off-diagonal covariance can push the effective count above the benchmark count, and then $K_{\mathrm{eff}} > K$ is possible. A single positive-variance benchmark has $\Lambda = 1$, although its off-diagonal coefficient is undefined because $K - 1 = 0$. This arithmetic count also differs from a spectral participation ratio.

A.5 Pooling and finite-sample bias

We pool T_c and U_c across configurations before taking their ratio, because each individual denominator uses two replicate degrees of freedom and can approach zero or become negative. Averaging individual ratios would instead give those unstable denominators disproportionate influence. Pooling reduces that problem but doesn't remove the effect of configurations with large covariance contributions.

Recomputing the recipe influences from the cluster linearisation gives participation ratios of 3.08 for margins and 9.02 for accuracy. We find that one recipe, dolma1.7-no-math-no-code, contributes 0.520 of the squared margin influence, and removing that recipe moves the estimate by 0.037, the largest change any single recipe produces. A leverage-corrected influence divides each recipe residual by one minus that recipe's share of the denominator sum, raising the share to 0.538 and lowering the two ratios to 2.92 and 8.56, with the concentration figure depending on the correction a reader assumes. These quantities describe influence concentration, and aren't literal counts of independent clusters for inference. Recomputing the permutation experiment ten times at the record's own 1,000 replicates gives margin coverage between 0.915 and 0.942 with a mean of 0.931. The recorded 0.925 sits inside that seed-to-seed spread. The record reads as one draw at a replicate count too coarse to separate these values, while the ten runs together put margin coverage 0.019 below nominal.

The covariance moments are unbiased under their assumptions, but the pooled ratio has a random denominator and its square root introduces a second nonlinear transformation. Thus neither of these two transformations inherits exact unbiasedness. A finite implementation must also specify what happens when a pooled sum is nonpositive. The reported data give finite full-battery estimates, though that observation doesn't establish finite estimates for arbitrary input populations.

A.6 Permutation reference

For each configuration and trait, we apply a uniform permutation π_j of run labels to both halves together. The diagonal products then sum to the same value, leaving U_c unchanged, and independent permutations across traits give zero expected off-diagonal products because each centred deviation sums to zero. Conditional on a positive fixed pooled denominator, this implies

$$\mathbb{E}_\pi \left[\frac{\sum_c T_c^\pi}{\sum_c U_c} \right] = 1. \tag{15}$$

This expectation for the squared ratio doesn't imply that $\mathbb{E}_\pi[\widehat{\Lambda}^\pi] = 1$. If all permuted ratios are nonnegative, concavity of the square root gives an expectation no larger than one, with strict inequality

for a nonconstant ratio. For example, equally likely ratios of 0.5 and 1.5 average to one, while their square roots average to about 0.966.

A common permutation across all traits preserves their pairings and can't remove cross-trait association. Independent trait permutations avoid that problem, and because the identity belongs in the uniform permutation sample space, its chance of reproducing the observation doesn't invalidate a permutation procedure. The construction preserves marginal score values and cluster membership, although it can alter dependence of run alignments across configurations. Validity as a test therefore requires an exchangeability assumption beyond zero mean covariance.

A.7 Planning calculations and information ratio

Our planning power calculation assumes approximately Gaussian half-score deviations. With $d^A = e + a$, $d^B = e + b$, independent errors a, b, and $\mathrm{Var}(a) = \mathrm{Var}(b) = k\,\mathrm{Var}(e)$, the product variance is

$$\mathrm{Var}(d^A d^B) = \sigma^4[(1+k)^2 + 1], \qquad k = 2(1-\rho_g)/\rho_g. \tag{16}$$

Here $\rho_g = \Lambda^2/(\Lambda^2 + \bar{v})$ is the aggregate reliability, with $\bar{v}$ the item-noise contribution on that scale. The planning values used two replicate degrees of freedom per configuration and design effect 1.60. Appendix B preserves the resulting values for the intended 85-configuration analysis. That intended analysis differs from the realised 125-configuration study, and we don't infer realised power from the observed effect.

For an infinitesimal additive shift in Gaussian item margins, the ratio of accuracy to margin noise per unit squared sensitivity is

$$\frac{p(1-p)}{\phi(z_0)^2}, \qquad z_0 = \Phi^{-1}(1-p). \tag{17}$$

At an illustrative accuracy of $p = 0.35$ this model gives about 1.66, and its minimum is $\pi/2$, which means the planned lower threshold of 1.4 for per-trait ratios couldn't falsify the model (prediction eight of the plan, Appendix B). The calculation describes only this location-shift model and shows no empirical advantage for other margin distributions.

For the information calculation, let item margins follow a Gaussian location model with standard deviation s_m. An infinitesimal shift δ changes accuracy by $f(0)\delta + o(\delta)$, while it changes the mean margin by δ. Under independent item sampling, the item-noise variances are $p(1-p)/n$ and s_m^2/n, with the Gaussian model giving $f(0) = \phi(z_0)/s_m$. Dividing noise by squared local sensitivity yields Equation 17, and both the scale cancellation and minimum $\pi/2$ require the Gaussian location model used in this calculation.

The competence and gain adjustments carry further measurement limitations. The opposite-half competence proxy uses estimated seed standard deviations, including a 10^{-6} floor where the margin variance estimate for WinoGrande is -3.15×10^{-7}. The gain covariate uses both halves, and that lets its item noise correlate with the outcome errors. The reported zero-mediation simulation treats this shared item component differently from the observed regression. Its margin reference begins at 1.24502 instead of the observed 1.24395, and moves to 1.0910 with standard deviation 0.096 against the fully adjusted observed value of 1.276. The accuracy reference begins at 1.11831, above the observed 1.07837, and ends at 1.1045 against an adjusted 1.094. These mismatches prevent a calibrated causal interpretation of the residualisation comparisons.

B Design details

B.1 Common step rule

For each of the 125 configurations, we selected the largest checkpoint step released for all three runs. The supplement described in the analysis record omits the fetch stage's complete configuration table. DataDecide stops the two auxiliary runs below 1B at 25 percent of the 1B compute budget and trains three complete runs only at 1B (Magnusson et al., 2025). The per-instance release contains auxiliary runs from 150M upward. The replicated sizes begin at 150M, and none of the smaller models has replicates.

An index of the 25 released recipe archives reproduces the selected step recorded for all 375 runs. At 530M the selected step averages 86.7 percent of the default run's final step, with configuration values from 38.9 to 88.7 percent. At 750M it averages 41.5 percent, with values from 17.7 to 44.0 percent. In 92 configurations the three runs have different maximum steps. In 26 of them, one at 530M and 25 at 750M, the largest maximum step exceeds the smallest by more than a factor of 1.5. Only 33 configurations share a final step, and 24 of those sit at 1B.

Table 4 recomputes both estimates on subsets defined by the index. The full sample gives margin inflation of 1.244 with interval 1.143 to 1.338 and accuracy inflation of 1.078 with interval 0.993 to 1.157. Without the 26 severely truncated configurations, margin inflation is 1.276 with interval 1.114 to 1.420 and 1.120 for accuracy, and neither wild interval includes one. The 92 configurations with different final steps give 1.330 with interval 1.024 to 1.583 and 1.070 with interval 0.956 to 1.171.

The 33 configurations with a shared final step give intervals too wide to separate inflation from one, and 18 of their 24 recipes contribute only a 1B configuration. A configuration's influence, $(T_c - \widehat{\Lambda}^2 U_c)/\sum_{c'} U_{c'}$, has Spearman correlation -0.036 for margins and 0.001 for accuracy with the selected step's share of the default final step. A permutation of that share within each size over 4,999 draws gives $p = 0.314$ and $p = 0.990$. Because we chose these subsets after we saw the full-sample estimates, they describe sensitivity and test no hypothesis. Table 6 gives estimates by size as a further check.

Table 4. Inflation estimates on subsets defined by each configuration's checkpoint schedule, with wild cluster bootstrap-t intervals over 4,999 draws. Severe truncation means that the largest final step among the three runs exceeds the smallest by more than a factor of 1.5.

Subset	Configurations	Recipes	Margin	Accuracy
All	125	25	1.244 [1.143, 1.338]	1.078 [0.993, 1.157]
Shared final step	33	24	1.082 [0.121, 1.530]	1.129 [0.769, 1.397]
Without severe truncation	99	25	1.276 [1.114, 1.420]	1.120 [1.020, 1.208]
Severe truncation only	26	25	1.107 [0.692, 1.397]	0.977 [0.357, 1.306]
Different final steps	92	25	1.330 [1.024, 1.583]	1.070 [0.956, 1.171]

B.2 Reliability by trait

The item halves contain 18,840 and 18,842 items across the battery, split within each trait and recorded for reuse across runs. The planning document proposed that we estimate reliability on eight screening recipes before we inspect seventeen estimation recipes. We didn't implement that separation, and Table 5 instead uses all 25 recipes. The plan predicted margin inflation above 1.349 and accuracy inflation above 1.40, which none of its $\binom{25}{17} = 1{,}081{,}575$ possible estimation sets reaches. The reliability checks and the decisions based on them remain exploratory.

The battery mean reliability is 0.324 on accuracy and 0.652 on margins, and BoolQ reaches 0.970 and 0.993 while ARC-Easy margins reach 0.927. Two accuracy traits have negative reliability estimates, and two others fall below 0.10. A recomputation of the per-trait information ratios gives a median of 1.608, a minimum of 1.571, and a maximum of 1.732, the same three figures the pipeline summary reported, although the summary didn't release the ratios themselves. The battery-level ρ_g recomputes to 0.961 on margins and 0.734 on accuracy, against planning values of 0.61 for margins and 0.49 for accuracy at $\Lambda^2 = 1.819$.

B.3 Power calculations

Table 3 of the extended supplement reproduces the planning arithmetic for 85 configurations and seventeen recipe clusters, whereas the completed analysis uses 125 configurations and 25 clusters. On the Λ scale, the reported linearised standard errors are 0.045 for margins and 0.038 for accuracy, compared with planning values of 0.0935 and 0.1232. The corresponding variance-ratio standard errors, 0.112 and 0.081, belong to the Λ^2 scale and aren't comparable with the planning column.

Table 5. Reliability of one item half's run deviations across all 25 recipes. Reliability equals $\widehat{\sigma}_E^2/(\widehat{\sigma}_E^2 + \widehat{\sigma}_{\text{noise}}^2)$ and can be negative when the estimated run component is negative. MC denotes a multiple-choice scoring format in this table.

Trait	Format	Accuracy	Margin
ARC-Easy	MC	0.602	0.927
ARC-Challenge	MC	0.203	0.584
BoolQ	yes/no	0.970	0.993
CommonsenseQA	MC	0.615	0.930
HellaSwag	cloze	0.450	0.757
MMLU	MC	0.022	0.860
OpenBookQA	MC	0.091	0.392
PIQA	cloze	−0.028	0.273
Social IQa	MC	0.416	0.888
WinoGrande	cloze	−0.102	−0.086
Battery mean		0.324	0.652

The realised standard errors are smaller by factors of 2.1 and 3.3, whereas the larger configuration count alone would predict a factor of approximately 1.21. Interpolation of the N1 simulation grid gives dispersion at the planned $\bar{r}_E = 0.0911$ that is 1.04 times the dispersion at the observed margin value of 0.061 and 1.13 times the dispersion at the accuracy value of 0.0181. The design effect at $\rho_{\text{ICC}} = 0.15$ and the reliability also differ, but we haven't separated their contributions. The planned accuracy MDE is 1.412, with a separate power entry at $\bar{r}_E = 0.115$. We retain the original planning calculations and don't recompute power from the estimated effect after we saw the results.

B.4 Decision checks and deviations

The planning document lists nine checks with proposed responses. Its claimed date of September 5, 2026 lacks an independent timestamp. We treat it as a reported plan and its checks as exploratory assessments.

A batch offset can enter the same cross-trait products as run-level variation. The auxiliary-run contrast removes it while the three-run estimate stays exposed. The batch-offset null N6 adds a shared default-versus-auxiliary offset with standard deviation 0.05 to the zero-correlation population and gives mean factors of 1.004 on margins and 1.003 on accuracy over 2,000 replicates, with 1.000 on both scales under the auxiliary-run contrast. Check G8 passes on point estimates, with an auxiliary-run margin of 1.256 against the gate's threshold of 1.122, which is one plus half the full-battery excess of 0.244, although the intervals cross the decision boundaries and the full-set estimate stays uncorrected for this exposure.

Under the default gain simulator, $\sigma_g = 0.05$ produces approximately 61 percent of the observed margin excess. An empirically matched simulator produces approximately eight percent at the same gain scale and six percent at the estimated scale of 0.042. These results describe sensitivity to simulator assumptions and aren't causal bounds on the contribution of gain. The accuracy result gives no independent protection because the multiplication of a margin by a positive factor leaves its sign unchanged. Table 4 of the extended supplement records every planning check beside its recorded outcome. Those outcomes include the 125 parsed cells at G0, power of 1.00 against an inflation of 1.427 at $\rho_g = 0.73$ for G1, a median information ratio of 1.61 at G2, and an offset of at most 0.002 at G4.

The analysis didn't execute the proposed responses for G3 and G5. G3 combines asymmetry with negative variance estimates, although a negative moment estimate doesn't establish leakage. The records lack a usable source-document identifier for the proposed alternative split. We retain the margin analysis after the G5 failure with the simulator sensitivity disclosed, which departs from the proposed response. The source gate table has seven rows, and transport and the auxiliary-run contrast supply G6 and G8.

B.5 Reported planning predictions

The planning document specifies eleven predictions with rejection conditions, and Table 5 of the extended supplement lists each criterion with its outcome, and we don't treat the document as verified preregistration. The first prediction requires $\Lambda_M > 1.349$ and $\Lambda_A > 1.40$ with lower confidence limits above one, and the second requires $\Lambda_A \geq 0.8\Lambda_M$. Predictions three and four bound the accuracy excess retained after gain adjustment, the N5 share of the margin excess, and the N6 share of the full-set excess. Predictions five to seven concern underprediction by independence across size bands, transport to PolyPythias, and an external seed mechanism with a bootstrap lower limit above one. Prediction eight requires at least eight traits with information ratios between 1.4 and 3.0, and prediction nine requires a cross-format diagnostic above 1.2 whose lower confidence limit also exceeds one. Prediction ten requires a Frobenius distance above 0.5, and prediction eleven more than ten percent of same-size configuration pairs with $P_{\text{flip}} > 0.05$.

B.6 Excluded populations

MultiBERTs provides 25 seeds for one configuration (Sellam et al., 2022), whereas the 100 BERT runs of McCoy et al. (2020) vary fine-tuning from one pretrained checkpoint. The Pythia size ladder itself lacks seed replicates (Biderman et al., 2023). The OLMo-2 checkpoints lack pretraining-seed replicates (Team OLMo et al., 2025a), since their midtraining soup ingredients share one stage-1 checkpoint and vary only data order. The signal-and-noise panel lacks per-item outputs, which limits its use to the external check. We exclude these populations because they lack the matched replicates or the item outputs required here.

B.7 Relation to prior measurement work

Reimers and Gurevych (2017) compare score distributions across seeds, Madaan et al. (2024) report benchmark-level seed variance, and Hagmann et al. (2023) model nondeterminism with scalar variance components. Our target adds covariance between benchmark-specific run deviations. Jordan (2024) compares shifted versions of one benchmark, and Bui et al. (2025) and Hochlehnert et al. (2025) vary fine-tuning and decoding seeds, a different source from pretraining-run variation. Dodge et al. (2020, Section 4.3) compare four GLUE task rankings across seeds without a covariance matrix.

The Spearman matrix of Zhou et al. (2020, Figure 3) averages per-seed correlations computed over training checkpoints, which measures co-movement along a trajectory, and seed deviations at a fixed step never enter it. Dufour et al. (2026) separate training from sampling variation, Miller (2024) covers item-sampling uncertainty with clustered and paired two-model comparisons with each evaluated model treated as fixed, and Zhao et al. (2026) set a three-seed noise floor on a twelve-task average and combine per-task Welch tests by Stouffer's method, and they note that positive task correlation would make the combination conservative but don't estimate that correlation. Haupt et al. (2026) estimate per-item variation on PolyPythias, a quantity distinct from the covariance of benchmark run effects.

Cheverud (2001), Nyholt (2004), and Li and Ji (2005) count effective independent tests to adjust significance thresholds, whereas we convert a noise covariance into an effective benchmark count. Searle et al. (1992) explain negative moment estimates when the true component lies near zero. Schaeffer et al. (2023) show how discontinuous metrics change apparent trends, and accuracy thresholds our margin at zero. We report both scales, although low seed signal relative to item noise limits accuracy here.

Dehghani et al. (2021), Demšar (2006), Dror et al. (2017), Card et al. (2020), Gorman and Bedrick (2019), and Burnell et al. (2023a) address benchmark selection, inference, power, split dependence, and per-instance reporting. Henderson et al. (2018) and Agarwal et al. (2021) resample runs within tasks, whereas we must preserve paired run effects across benchmarks. Hardy et al. (2026), Kohli (2026), and Sha and Zhao (2026) measure dependence across models or judges, and Messing (2026) decomposes variance from prompt and judge choice, whereas we measure paired deviations across training runs.

The cross-half identity follows the repeated-measurement correction of Spearman (1904) and the split-half construction of Spearman (1910) and Brown (1910), and Cronbach (1951) motivates the

50 re-splits in N3, which change the margin estimate by a reported standard deviation of 0.002. The Spearman-Brown formula shares the form $1+(k-1)\rho$ with Λ^2, but its correlation concerns parallel items. The reliability means in Table 5 don't estimate our inflation. Campbell and Fiske (1959) cross traits with methods, whereas no benchmark here appears in multiple formats, and content and scoring format therefore can't be separated.

Brennan (2001) provides the variance-component framework for these distinctions. Hofmann et al. (2025) and Kipnis et al. (2025) describe ability structures across models, a different object from within-configuration noise covariance. Hedges et al. (2010) handle dependent effect sizes without a complete covariance matrix, whereas we estimate that covariance.

C SUPPLEMENTARY RESULTS

C.1 Covariance diagnostics The largest discrepancy between opposite cross-half covariance entries is 2.44 bootstrap standard errors, a size that doesn't by itself establish leakage. The reduced item records provide 57 MMLU subject groups but only one group for each other trait. The passage-aware split in Appendix A uses the release's request files instead.

Among the eight accuracy traits with positive estimated seed variance, estimated correlations range from -0.216 to 1.296. The symmetrised matrix has a negative eigenvalue of -0.55. These estimates don't form a valid correlation matrix, a possible finite-sample outcome of noise-corrected moment estimation. We retain the unprojected matrix because a positive-semidefinite projection would change the reported values without an established correction target, and the primary ratio uses cross-half sums directly. The nine-trait margin matrix has nine positive eigenvalues, whereas the indefinite accuracy matrix yields no validated count of independent population directions.

The original scorecard counted five failures, four passes, and two predictions that couldn't be tested as written. Predictions 9 and 11 came from a separate analysis of the per-run reductions, outside the original pipeline. Prediction 7 lacks its required lower confidence limit, and prediction 6 depends on a post hoc choice to compare excess inflation. Table 5 of the extended supplement retains the numerical outcomes and states where the assessment differs from the proposed criterion.

C.2 Variation across model sizes Each size band contains 25 configurations, one per recipe, and its clusters are therefore singletons without within-band replication. Only the 530M interval excludes one on either score scale, as Table 6 reports. The margin exclusion disappears under the test-inversion interval reported below. The intervals don't support a ranking of model sizes. All five margin and four accuracy point estimates exceed one. Prediction 5 passes a point-estimate reading and fails the stricter interval reading of the original scorecard.

Table 6. Inflation estimates by model size across the 25 recipes. The intervals use the wild cluster bootstrap-t calculation described in Appendix D.

Size	Λ_M	95% interval	Λ_A	95% interval
150M	1.101	[0.781, 1.344]	1.028	[0.878, 1.165]
300M	1.350	[0.638, 1.803]	1.080	[0.836, 1.288]
530M	1.514	[1.064, 1.837]	1.219	[1.083, 1.338]
750M	1.110	[0.686, 1.407]	0.987	[0.310, 1.320]
1B	1.149	[n/a, 1.751]	1.205	[0.805, 1.497]
Pooled, $N = 125$	1.244	[1.143, 1.338]	1.078	[0.993, 1.157]

The lower squared-ratio bound for 1B margins falls below zero and leaves its square-root endpoint unavailable. Three positive constructions give finite endpoints, and none of them excludes one. The cluster-robust $t(24)$ interval on the log of the squared ratio gives 0.811 to 1.628. The configuration percentile bootstrap gives 0.834 to 1.643. The wild bootstrap on the log of the squared ratio gives 0.491 to 1.656.

That same run also estimated every size band with BoolQ removed. On margins the bands give 1.548, 1.864, 1.874, 1.888 and 1.649 from 150M to 1B, with intervals of $[1.325, 1.742]$, $[1.763, 1.961]$, $[1.694, 2.039]$, $[1.660, 2.082]$ and $[1.215, 2.001]$. Every band then excludes one,

and the 1B cell recovers a finite lower endpoint. On accuracy the same removal gives 1.842, 1.900, 1.548, 1.594 and 1.222, with intervals of $[0.989, 2.408]$, $[1.174, 2.417]$, $[1.262, 1.792]$, $[1.341, 1.811]$ and $[0.429, 1.674]$. The middle three exclude one while the smallest and largest bands don't. We read this as the composition sensitivity that reappears at size resolution, and not as independent support for the pooled estimate, because the full-battery sweep already names BoolQ as the benchmark that pulls hardest toward one.

A separate run scored those constructions inside one size band at the three runs the release ships, and the transform doesn't repair that case. On an accuracy-like band the log interval misses in 0.151 of replicates against 0.017 for the plain one. It buys definedness on that band at a cost in coverage. The leverage correction repairs more, and it lifts the margin band to 0.942 at three runs and the accuracy band to 0.952 at ten.

The 1B log-scale endpoints should therefore be read as closer to a ninety percent statement than a ninety five percent one (seed 20260972, 4,000 replicates). At the full 125-cell design the two constructions agree within 0.006 across the dependence structures we simulated, and both fail together near 0.82 at a band-shared quarter. The leverage correction alone improves on either and holds 0.954 at the joint corner. The transform keeps endpoints positive in a thin cell but doesn't give a better interval (seed 20260978, 4,000 replicates).

Simulation says these per-size rows should be read as estimates with wide uncertainty, since they don't carry their nominal level as intervals. The accuracy column is the weaker of the two. One band at three runs covers 0.930 on a margin-like population and only 0.707 on an accuracy-like one, and ten runs lift the accuracy case to 0.947. Power inside a single band is worse again, since at a true margin ratio of 1.2446 a single band excludes one in 0.313 of replicates at three runs, 0.739 at six and 0.957 at ten. At a true accuracy ratio of 1.078 the exclusion rate reaches only 0.081 at ten runs and 0.317 at forty. No affordable run count settles the accuracy question inside one band, which is why the paper draws its conclusions from the pooled design (4,000 replicates, seeds 20260963 and 20260971).

Figure 1 of the extended supplement reports N1 calibration at six effect sizes with 2,000 replicates each, using the 2.5th and 97.5th percentiles of estimates as dispersion bands. Table 9 of the extended supplement gives those recovered estimates alongside the other null and sensitivity runs. The bands describe simulated estimator dispersion and say nothing about confidence-interval coverage.

C.3 Permutation comparison N2 was proposed as an estimator-bias check and was never the primary inferential procedure. In the reported 500 permutations, none reaches the observed margin estimate, and 0.028 reach the accuracy estimate. The observed configurations are held fixed under this reassignment, while trait-specific run labels are permuted independently. The primary bootstrap instead targets variation across recipe clusters. The two procedures use different randomisation and sampling assumptions, which is why we report both accuracy results.

C.4 Associations with scoring format The cross-format diagnostic retains the diagonal sum and only those off-diagonal terms whose traits have different formats. It has a null reference of one when those retained covariances vanish, but the masked sum needn't represent the variance of a realisable benchmark average. Because each benchmark has only one format, the comparison can't identify a causal effect of format.

The separate calculation reproduces the full estimate to four decimal places and then retains 54 of the 90 ordered off-diagonal pairs. It gives 0.921 with interval 0.801 to 1.035 for margins, and 0.965 with interval 0.848 to 1.063 for accuracy, both below the proposed 1.2 threshold. The margin result falls below the 1.153 expected from uniformly distributed covariance contributions under the comparison used here, and below the 1.113 that a uniform shared item component with the whole margin excess alone, with no seed covariance, would imply, a value the interval also excludes (Appendix C.15).

All of BoolQ's off-diagonal pairs cross a format boundary, since it is the only yes/no benchmark. Its margin seed variance is 1.43×10^{-3}, compared with 4.34×10^{-4} for the next-largest trait. Without BoolQ, the cross-format margin diagnostic is 1.006 with interval 0.972 to 1.039, compared with an unrestricted nine-trait estimate of 1.786 with interval 1.689 to 1.877.

Among the nine traits with defined margin correlations, without WinoGrande, the mean correlation is 0.624 across sixteen within-format pairs and 0.007 across twenty cross-format pairs. The three

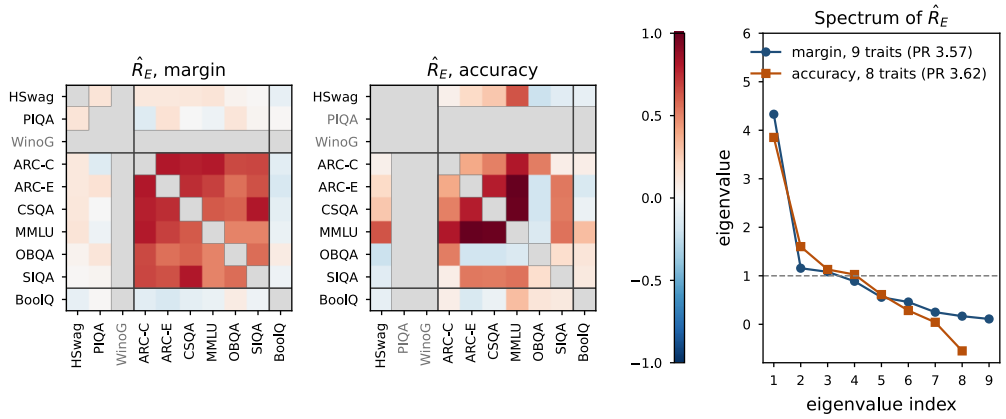


Figure 3. Cross-half correlation estimates across the 125 configurations, margins on the left and accuracy in the centre, with undefined variances shown in grey and eigenvalue spectra on the right. The accuracy estimate has a negative eigenvalue, and its spectrum is a diagnostic of the moment estimate and isn't a valid population correlation spectrum. PR is the spectral participation ratio, 3.57 on the nine margin traits and 3.62 on the eight accuracy traits after clipping that eigenvalue.

cloze traits give margin inflation of 1.225 with interval 1.064 to 1.367. Without ARC-Easy and ARC-Challenge in the multiple-choice block, inflation is 1.479 with interval 1.416 to 1.541. These associations show concentration within the larger format groups. Benchmark content and format, however, remain confounded in this battery. A rank-one model with trait-specific loadings can also produce uneven contributions across those groups.

C.5 A two-correlation fit We replaced every off-diagonal correlation in the margin matrix with 0.624 for pairs sharing a format and 0.007 for pairs that don't, and kept each benchmark's estimated seed variance. WinoGrande enters with zero weight, since its variance estimate isn't positive. The fitted matrix gives inflation of 1.285 for the full battery against the observed 1.244. Without BoolQ it gives 1.728 against 1.786, and for the four multiple-choice benchmarks other than the two ARC sets it gives 1.456 against 1.479. The three cloze traits are the weakest case, at 1.274 against 1.225, because PIQA and HellaSwag carry seed variances near 10^{-6} and WinoGrande none.

Equal variances with the same two correlations give 1.803, and that shows how much of the gap between the correlation structure and the observed factor comes from the variance weights. For the leave-one-out check we re-estimated both block correlations without the removed benchmark, which moved them between 0.572 and 0.657 within formats and between -0.022 and 0.050 across them, and then predicted inflation for the remaining nine. Those predictions differ from the observed deletions by between 0.024 and 0.054 in absolute value, and they overstate inflation in every deletion except BoolQ's. Both correlations come from the same runs as the observed values. This check tests the compression of the matrix and doesn't replace an out-of-sample test on a new battery.

C.6 Scoring-scale interventions We transformed the saved margins item by item and recomputed each estimate with the items, runs, split, and interval routines held fixed. For benchmark j, s_j is the pooled standard deviation of its per-item margins, which ranges from 0.087 for WinoGrande to 0.468 for CommonsenseQA. Table 7 of the extended supplement reports these estimates, and a tenfold rescaling of every margin leaves the estimate unchanged to nine decimal places, a check on the construction.

Division by s_j changes only the benchmark weights and lowers margin inflation from 1.244 to 1.182. The three bounded scores lower it further to between 1.120 and 1.141. The sign score gives 1.079 with interval $[0.992, 1.158]$, which includes one and differs from the accuracy estimate of 1.078 by 0.0007, a gap that only items whose stored margin is zero, from exact ties or float16 underflow, can produce, since elsewhere the sign score is an affine map of correctness and the tenfold-rescaling check shows the estimate is scale-invariant. Across these scores the off-diagonal sum falls from 0.547 to 0.164 times the trace while BoolQ's share of that trace grows from 0.679 to 0.844, and

bounded margins from the same runs reproduce between 62 and 75 percent of the gap between margin and accuracy inflation. These transformations change the scoring rule and leave the trained models untouched. They don't show how training produces large margins, and the intervals treat each s_j as fixed.

Table 8 of the extended supplement applies $\bar{r}_E = 0.0181$ with trait standard deviations held fixed, with flip probabilities interpreted under the stated Gaussian comparison model. The held-out accuracy point errors favour the full matrix over independence, 0.0105 versus 0.0143, without an interval for their difference. Every predictor also receives held-out seed standard deviations unavailable without replication. These results don't establish an operational replacement for replicated runs. The three-unit PolyPythias accuracy estimate of 2.286 differs from 1.078 and has a configuration-bootstrap interval with no defined lower endpoint, which further limits transport of this calculation.

C.7 Performance correlation as a noise proxy The reported Frobenius distances between R_E and R_P are 5.59 for margins and 6.49 for accuracy. Negative estimated variances leave eighteen margin and 34 accuracy ordered off-diagonal correlations undefined. The implementation fills these entries with zero before it takes differences, and that includes undefined diagonal entries compared with unit diagonals in R_P. Those diagonal substitutions contribute 1.0 and 2.0 to the squared distances.

A repair of only those diagonals gives distances of 5.499 and 6.339. If we restrict the comparison to jointly defined entries, the distances are 4.374 and 4.593, still above the planned threshold of 0.5. The threshold alone discriminates little between ten-trait matrices. The entrywise comparisons provide a more useful description.

The off-diagonal association between the two matrices is 0.756 for margins and 0.345 for accuracy. Mean defined R_E entries are 0.281 and 0.273, compared with all-pair R_P means of 0.750 and 0.754, and on matching pairs the R_P means are 0.743 and 0.769. The all-pair comparison therefore differs in scale by factors of approximately 2.7 and 2.8. The unadjusted proxy predicts a mean margin aggregate standard deviation of 0.0072, compared with the reported value of 0.0057.

An additional exploratory baseline, P2R, rescales R_P off-diagonals by the training-fold ratio of mean noise correlation to mean performance correlation. Its mean factors are 0.366 for margins and 0.370 for accuracy, compared with whole-population ratios of 0.375 and 0.362. On the same five recipe folds, its squared prediction errors are 0.0131 and 0.0342. It improves on P2 but remains 2.3 times worse than P1 on margins and worse than independence on accuracy. Its direction relative to independence holds on margins in all 1,000 random fold partitions and on accuracy in 987 of them, but the recipe-bootstrap intervals below include zero on both scales. This rescaling also requires replicated runs to estimate its factor.

C.8 Paired uncertainty for the prediction comparison Table 7 gives the fold-mean prediction errors that Section 4.5 summarises for each of the six models. Each model receives the held-out marginal seed standard deviations, and this design thus isolates the correlation component. On margins, the rank-one, rank-one plus gain, and full models each lower the mean squared error of log aggregate standard deviation by about 0.03 from the 0.0358 of independence, with paired recipe-bootstrap intervals that exclude zero, although the rank-one interval reaches zero when nonpositive held-out variances are filled from training folds, and on accuracy every interval for a difference from independence includes zero. The unrescaled phenotypic plug-in has the largest error on both scales, and in an operational comparison that refits every matrix without held-out marginals and predicts each held-out configuration's raw variance moment, no model separates from independence. Before we fit any model we clip negative variance estimates, while the target retains the original covariance contributions.

The rank-one model has the lowest margin point error and full covariance the lowest accuracy point error. Paired recipe-bootstrap intervals with 2,000 refitted draws separate neither from its runner-up. The unrescaled phenotypic plug-in has the largest error on both scales, and rescaling its off-diagonals on training folds reduces that error but needs replicate information and still trails the best model.

A rerun with every prediction, target, and squared log error saved reproduces all twelve fold means in Table 7 exactly, with P2R factors of 0.366 and 0.370. Before we computed any contrast we fixed eight paired differences and a practically meaningful difference of 0.005 in mean squared log error,

Table 7. Mean squared prediction error of log aggregate standard deviation across five recipe folds, with paired recipe-bootstrap intervals for their differences given below. The rescaled plug-in is an exploratory addition to the original five-model comparison.

Model	Margin	Accuracy
Independence	0.0358	0.0143
Rank-one competence	0.0057	0.0140
Rank-one plus gain	0.0063	0.0260
Phenotypic correlation plug-in	0.1073	0.1682
Full covariance	0.0068	0.0105
Rescaled phenotypic plug-in	0.0131	0.0342

corresponding to a 7 percent error in the aggregate standard deviation. The intervals resample the 25 recipes 2,000 times and refit the whole comparison in each draw, with the designed recipes treated as exchangeable clusters, and that absorbs fold-partition variation.

On margins, P1, P1G, and P3 each lower the error relative to independence by about 0.03, with intervals of -0.058 to -0.001, -0.059 to -0.002, and -0.060 to -0.004. Between 95 and 97 percent of draws put the reduction beyond 0.005. However, the P1 interval reaches zero when nonpositive held-out variances are filled from training folds. The P1 minus P3 interval runs from -0.003 to 0.012. These data don't separate the rank-one model from the full matrix. On accuracy, each of the four intervals for a difference from independence includes zero, and the interval for the full matrix minus independence runs from -0.017 to 0.023.

An operational comparison instead predicts each held-out configuration's raw aggregate variance moment from matrices refitted without held-out marginals. There the full, rank-one, and shrinkage models differ from independence on both scales by amounts whose 2,000-draw recipe-resampling intervals all include zero. That target is the noisy three-run moment, so the comparison bounds detectable gains but doesn't calibrate the ranking above.

The 1,500 same-size pairs of the flip census also have seed-matched runs. The paired difference carries a positive variance estimate for 1,445 margin and 1,222 accuracy pairs. For those pairs, the ratio of the paired-difference standard deviation to the independence value has median 1.024 on margins and 1.081 on accuracy. Its 5th to 95th percentiles span 0.39 to 1.40 and 0.36 to 1.62. The stricter subset with positive variance in both configurations gives medians of 1.027 and 1.079 over 1,379 and 919 pairs. Across the full 1,500-pair census the seed-matched covariance is positive in 0.49 of pairs.

Figure 2 of the extended supplement draws all eight contrasts with both sources of uncertainty side by side. Across 1,000 random five-fold partitions of the original recipes, independence never has the lowest margin error, and P3 leads on accuracy in 55 percent of partitions against 39 for independence. The partition behind the table understates that margin advantage, since its independence error of 0.0358 lies below the partition median of 0.0438 and its P1 error of 0.0057 above the median of 0.0031. These contrasts concern correlation prediction with held-out marginal variances supplied. They don't carry over to the operational comparison in the main text.

We also ran the comparison on simulated populations with known seed covariance, with 300 replicates for each structure in Table 9. There we scored every model against the noisy fold target and the true aggregate standard deviation. The noisy target itself has a mean squared log error against the truth of 0.011 to 0.012 on margins and 0.024 to 0.032 on accuracy. With the matched margin covariance, P1, P1G, and P3 beat independence by 0.043 to 0.046 against the noisy target and by 0.048 to 0.051 against the truth.

With independent seed deviations, independence has the lowest margin error under both targets, and P1, P1G, and P3 lose between 0.001 and 0.007. On accuracy with the matched covariance, the true errors of P0, P1, P1G, and P3 differ by at most 0.0032, too little for a five-fold comparison to separate beside the target noise. These populations share one covariance across recipes with little systematic variation between configurations. They don't reproduce how the performance-correlation plug-ins behave. We don't use them to assess P2 or P2R.

C.9 Recipe, size, and trait sensitivity Removal of each recipe in turn gives margin estimates from 1.207 to 1.256 and accuracy estimates from 1.065 to 1.099. The maximum changes from the complete battery are 0.037 and 0.021. Removal of each size gives ranges of 1.178 to 1.289 and 1.047 to 1.115, compared with individual-band ranges of 1.101 to 1.514 and 0.987 to 1.219. These overlapping intervals don't establish a size trend in the effect.

Trait removal changes the estimates more than recipe or size removal. Exclusion of BoolQ gives 1.786 with interval 1.689 to 1.877 for margins, and 1.558 with interval 1.412 to 1.687 for accuracy. Exclusion of CommonsenseQA gives the smallest point estimates, 1.096 and 1.014. BoolQ contributes 68 percent of the estimated margin variance trace and 84 percent of the accuracy trace.

The removal estimates imply that the off-diagonal terms with BoolQ are negative and sum to about -0.15 of the margin trace and -0.06 of the accuracy trace. Its removal therefore lowers the denominator by more than the numerator. The same removal lowers the aggregate standard deviation on both scales, from 0.00572 to 0.00516 on margins and from 0.00514 to 0.00326 on accuracy, as Table 11 of the extended supplement shows.

A larger inflation factor accompanies a less variable average. WinoGrande is the only other removal whose accuracy interval, 1.023 to 1.178, excludes one. The table gives the remaining intervals and standard deviations. We also lowered BoolQ's weight below the flat tenth in seven steps to 0.0533 and raised the other nine weights together so the battery still sums to one. Margin inflation climbs from 1.2439 at the flat weighting to 1.5030 at the smallest weight, and every interval along the way excludes one.

Accuracy climbs from 1.0784 to 1.2311 over the same steps. Its interval excludes one at a BoolQ weight of 0.0867 and below, a reduction of about one seventh. The estimate moves smoothly, so the accuracy conclusion is a knife edge in the weighting, since a battery that discounts BoolQ by a seventh reports dependence on accuracy as well. We report the flat weighting because the release gives no basis for a special BoolQ weight. The item-count weighting below is the one alternative its own counts supply. A reader with such a basis should expect the accuracy answer to move first.

Standardisation of traits by estimated seed standard deviation gives 1.803 on nine margin traits and 1.707 on eight accuracy traits. These point diagnostics omit traits with nonpositive variance estimates and divide by uncertain standard deviations, with seven accuracy traits at or below item noise. With those estimated weights held fixed, the wild recipe-cluster intervals are 1.544 to 2.023 and 1.228 to 2.076. They omit the uncertainty in the weights themselves.

We don't score them against the planning thresholds. Removal of defined traits from the standardised batteries gives margin estimates from 1.625 to 1.910 and accuracy estimates from 1.388 to 1.780. Selection of traits by positive estimated variance can change the target, and it could raise the mean correlation if the excluded low-signal traits also have low true correlations, an assumption we haven't tested.

The raw trait-removal spans are 0.690 for margins and 0.544 for accuracy, against interval widths of 0.195 and 0.164. The delete-one-trait jackknife formula gives scales of 0.554 and 0.455, against sampling standard errors of 0.045 and 0.038. Without the BoolQ removal value those scales fall to 0.139 and 0.073. These are descriptive composition diagnostics, because the ten fixed benchmarks aren't established exchangeable draws from a benchmark population.

A normal calculation on the jackknife scales and sampling standard errors added in quadrature gives ranges of 0.15 to 2.33 and 0.18 to 1.974, and both include one. Without the BoolQ value the same arithmetic gives 0.958 to 1.529 and 0.918 to 1.239. Those ranges aren't validated confidence intervals, since they can double-count sampling variation and the normal approximation behaves poorly near the parameter boundary, where $\Lambda = 0.15$ corresponds to $\bar{r}_E = -0.109$, only about 0.003 above $-1/9$. We retain the fixed-battery target and report composition sensitivity, and we don't treat these scales as an identified decomposition of total uncertainty.

A change in trait count also changes Λ at fixed effective covariance coefficient. At $\bar{r}_E = 0.061$, the nine-trait reference is 1.219 against the ten-trait 1.244. The corresponding accuracy values are 1.070 and 1.078. Four margin removals and five accuracy removals exceed those mechanical references. On eight traits, standardised accuracy inflation of 1.707 implies 2.75 effective benchmarks, while the raw accuracy effective coefficient of 0.0181 implies 7.10 effective benchmarks. The comparison of 2.75 with 7.10 holds trait count fixed, unlike comparison with the ten-trait value of 8.60.

The ARC pair has an estimated margin correlation of 0.801. Removal of both benchmarks gives margin inflation of 1.118 with interval 1.059 to 1.174, and accuracy inflation of 1.012 with interval 0.929 to 1.086. The pair affects the estimate, although the remaining margin interval still excludes one.

C.10 Benchmark count, identity, and weights To separate benchmark count from benchmark identity, we computed both estimates for all 1,013 equal-weight subsets of two to ten benchmarks, each with a wild recipe-cluster interval over 4,999 draws. Table 12 of the extended supplement summarises these subsets separately for each size K. Median margin inflation lies between 1.012 and 1.244 across sizes. The share of subsets whose interval lies above one rises from 38 percent at two benchmarks to 98 percent at eight. Accuracy medians stay between 1.017 and 1.078, and at each size from two to nine benchmarks between 20 and 31 percent of accuracy intervals lie above one. Among subsets of two to five benchmarks, 16 accuracy subsets have a nonpositive independence variance or a negative estimate of squared inflation, and so no defined estimate, and the table's quantiles omit them.

Subsets probe only the corners of the weighting. We also rebuilt the statistic under arbitrary weights and swept the interior. The rebuild reproduces the shipped values to 10^{-19} under flat weights, the check that licenses the sweep. Over 2,000 weightings drawn from a flat Dirichlet, 0.817 of margin intervals still exclude one. The estimate runs from 0.976 to 1.991 with a 5th percentile of 0.999. Over 2,000 weightings drawn from a Dirichlet concentrated near the flat point, every margin interval excludes one and the lowest estimate is 1.059.

Accuracy behaves in the direction that favours the reported claim, because 0.386 of flat-Dirichlet intervals and 0.402 of near-flat ones exclude one while the flat weighting itself gives 0.993 to 1.157 and doesn't. All but 4 of the 772 flat-Dirichlet exclusions and all 804 near-flat ones lie above one. The 0.018 of flat draws whose interval has an undefined endpoint count as draws that don't exclude it. The weighting we shipped is therefore the conservative one on accuracy. The conclusion on margins holds in 0.817 of flat-Dirichlet weightings and in all 2,000 near-flat ones. A rerun of the sweep from its committed script reproduces each of these figures to the third decimal.

At fixed benchmark count, subsets that contain BoolQ have mean margin inflation lower by 0.115 at two benchmarks and by 0.596 at nine. The corresponding accuracy differences are 0.076 and 0.499. At nine benchmarks every other benchmark raises mean margin inflation, by amounts from 0.004 for WinoGrande to 0.171 for CommonsenseQA. Across all sizes, margin estimates have a median of 1.136 with quartiles 1.018 and 1.390, and they run from 0.927 for HellaSwag with WinoGrande up to 1.786 in subsets that drop BoolQ and WinoGrande.

Accuracy estimates have a median of 1.038 with quartiles 1.001 and 1.229, and they run from 0.315 to 5.439. Two thirds of margin intervals exclude one, always from above, against 0.305 of accuracy intervals. A separate implementation recomputed every subset and reproduced each size's median and share, together with the full-battery 1.244 with 1.143 to 1.338 and 1.078 with 0.993 to 1.157. Since the subsets share runs, items, and configurations, their spread describes this battery and doesn't sample a wider population of batteries.

Item-count weighting changes the weights of all ten benchmarks and doesn't select on scores. It gives margin inflation of 1.119 with interval 1.068 to 1.170 and accuracy inflation of 1.043 with interval 0.963 to 1.119, while MMLU and HellaSwag carry 64 percent of the weight. A second reweighting, by the inverse pooled seed standard deviation, reproduces the standardised diagnostic above, which has to drop the benchmarks with nonpositive variance estimates. The deletion of recipes instead, two of the 25 at a time, leaves a margin interval above one in all 300 pairs, with estimates from 1.177 to 1.263 and a lowest lower limit of 1.037.

C.11 Interval construction and cluster sensitivity A percentile bootstrap over 125 configurations gives intervals of 1.069 to 1.426 for margins and 0.976 to 1.182 for accuracy. The margin interval excludes one, and the accuracy interval includes one. A comparison of squared linearised standard errors, with 0.045 against 0.093 on margins and 0.038 against 0.052 on accuracy, gives ratios of 0.23 and 0.53. The implementation separately reports numerator-based design effects (Kish, 1965) of 2.06 and 1.26 from $1 + 4\rho_{\mathrm{ICC}}$, compared with the planned 1.60.

These ratios describe different quantities and aren't contradictory estimates of one design effect, because the numerator calculation uses T_c, while linearisation uses $T_c - \widehat{\Lambda}^2 U_c$. The interval procedures also change resampling unit, studentisation, and denominator treatment together. We consequently don't infer a single intraclass correlation from their relative widths. We use the bootstrap-t (Cameron et al., 2008) because sandwich variances are biased with few clusters (Bell and McCaffrey, 2002), although it doesn't guarantee coverage at 25 clusters.

C.12 Observed gaps and predicted rank reversals With accuracy noise $\sigma_{\mathrm{agg}} = 0.00514$, the Gaussian calculation gives $P_{\mathrm{flip}} > 0.05$ for a single-run comparison with a true gap below 0.01196, and 455 of the $5\binom{25}{2} = 1{,}500$ observed same-size pairs fall below it. At a true gap of 0.005 the same single-run calculation gives a sign-reversal probability of 0.246, against 0.229 under independence. Since observed gaps are noisy, this count isn't a lower bound on the fraction of true gaps below the threshold and doesn't measure realised reversal frequency.

We also simulated a practitioner's correction under three-run noise, over 4,000 replicates that build the shipped design and estimate the ratio from it. Each replicate then tests a fresh comparison at 1.96 standard errors under an uncorrected rule, a plug-in rule scaled by the estimate, and an oracle rule with the true value. At a true margin gap of zero the rules declare in 0.113, 0.051 and 0.050 of replicates against a nominal 0.05, and three runs per model leave those at 0.115, 0.053 and 0.051. The plug-in rule sits within 0.005 of the oracle in every cell despite the estimate's noise. An accuracy-like population gives 0.067, 0.052 and 0.047.

Sweeps to ten, 80 and 360 runs per model show that extra runs buy nothing against the size error, since both standard errors fall with the square root of the run count. Across one to ten runs the uncorrected rule declares at a zero margin gap in 0.112 to 0.122 of replicates against 0.047 to 0.056 corrected, and accuracy gives 0.061 to 0.078 against 0.047 to 0.063. Runs buy power instead, from 0.100 of replicates at one run to 0.614 at ten for a margin gap of one aggregate standard deviation, or 0.111 to 0.615 on accuracy. Four fifths arrives only at 16 runs for a margin gap of one aggregate standard deviation, against three shipped, and at 20 on accuracy, where even the oracle misses it at 16 by 0.002.

Both sweeps estimate the correction on the source population. We also swept its transfer to targets whose true ratio differs, with 8,000 replicates per target. A correction from the shipped margin design, whose true ratio is 1.245, holds nominal size at 0.053 only at a target of 1.245. Below that it turns conservative, at 0.015 of replicates for a target of 1.00 and 0.028 at 1.10, and pays for that in power.

Above it the size error returns, at 0.080 for a target of 1.40 and 0.101 at 1.50 against 0.159 and 0.188 uncorrected, while a matched rule re-estimated on each target holds between 0.048 and 0.053. The accuracy design as source, with true ratio 1.078, gives the same shape and holds size at 0.056 for a target of 1.10 and 0.168 at 1.50. The transferred rule holds between 0.04 and 0.06 only within about 0.05 of the source ratio. Readers should measure the ratio on their own battery.

C.13 Adjacent-checkpoint variation in PolyPythias We compare all nine 160M PolyPythias runs at steps 143000 and 142000 on a nested 1,800-item sample. Nine non-MMLU traits are included, with items, scoring code, and batching held fixed. MMLU's 57 subjects would increase the forward passes approximately threefold to include one additional trait, which is why this check excludes it.

The margin battery's between-run standard deviation is 0.00513, while the root-mean-square checkpoint shift is 0.00456, a ratio of 0.890, and accuracy gives 0.00451 and 0.00710, a ratio of 1.573. Across the nine runs, final and preceding checkpoint scores correlate 0.573 for margins and -0.070 for accuracy, and our repeat rescore reproduces each of these values. Accuracy rises by 0.0041 on average between the two steps. Without that shared change the accuracy ratio falls to 1.279, while the margin ratio stays at 0.890. These results show substantial checkpoint sensitivity in this restricted sample. The design can't fully separate stochastic trajectory variation from systematic progress across 1,000 steps.

The estimator describes run variation at the scored checkpoint, which can include trajectory effects correlated across traits. It doesn't isolate initialisation from these effects in the available design. This comparison uses 1,800 items instead of 4,755, nine runs, and a final checkpoint, whereas DataDecide

uses common steps during training. On margins, BoolQ alone has between-run standard deviation 0.0438 and root-mean-square shift 0.0370 in this check. We didn't compute these values for the other benchmarks. The next paragraph measures the corresponding checkpoint sensitivity in DataDecide.

C.14 Adjacent-checkpoint variation in DataDecide We reduced every DataDecide run a second time at the largest checkpoint step that all three runs in its configuration share below the selected step. The gap is 1,250 steps at 530M and 750M, between 657 and 1,250 steps at 150M and 300M, and 1,869 steps at 1B. Two configurations, dclm-baseline-75p-dolma1.7-25p at 750M and dolma1.7-no-math-no-code at 1B, share no released step below the selected one. The comparison covers 369 runs in 123 configurations. Table 13 of the extended supplement reports both estimates at the two steps with the same items, split, weights, and interval routines.

Margin inflation falls from 1.233 to 1.192 at the earlier step. A paired recipe bootstrap over 4,999 draws gives a 95 percent interval of -0.078 to 0.001 for the change. Accuracy inflation rises by 0.048, from 1.082 to 1.130, with a change interval of -0.022 to 0.130. Neither change interval excludes zero, but the accuracy interval at the earlier step, $[1.044,\ 1.210]$, excludes one.

The reading of the full-battery accuracy estimate as close to independence depends on the scored checkpoint. A rerun with the seed-noise estimator on the same 123 configurations reproduces all four estimates. The centred test-inversion interval reported below gives 1.059 to 1.222 for accuracy at the earlier step, so that exclusion of one doesn't depend on the wild construction. The same inversion gives 0.996 to 1.169 for accuracy at the selected step and lowers the margin lower endpoints to 1.099 and 1.080. BoolQ's share of the accuracy trace moves only from 0.847 to 0.872 between the two steps.

In the form used for PolyPythias, the root-mean-square change in a run's aggregate score equals 0.469 between-seed standard deviations on margins and 0.631 on accuracy. The ratios are 0.493 and 0.618 after we remove each configuration's mean change. The movement differs between seeds and doesn't reflect shared training progress alone. Within configurations, the two steps' seed deviations correlate 0.870 for margins and 0.811 for accuracy.

The 750M band, where the selected step lies furthest from the final step, gives ratios of 0.889 and 1.048. Its margin ratio sits within 0.001 of the PolyPythias value of 0.890, while its accuracy ratio stays below the PolyPythias value of 1.573. The DataDecide gap there is 1,250 steps against 1,000 for PolyPythias. We added this comparison after we saw the full-sample estimates. It is exploratory like the rest of the empirical study.

C.15 Simulation and permutation calibration Under independent noise our calibration simulation returns mean inflation of 1.0009 on margins and 1.0004 on accuracy. The recovery slope across effective covariance coefficients from zero to 0.50 is 0.998, and fifty item re-splits on the observed data give standard deviations of 0.002 and 0.008. These checks assess selected reference distributions and can't exclude training-schedule confounding.

The linearised margin standard error of 0.045 is close to the N1 null dispersion of 0.046, while accuracy gives 0.038 against the simulated 0.048. The accuracy simulation dispersion gives a descriptive interval of 0.979 to 1.178, which also includes one, though it isn't a calibrated alternative because the simulated and observed populations differ.

A separate implementation of the interval definitions gives empirical coverage from 0.920 to 0.975 across sixteen cells of 200 datasets. The four cells with dispersion 0.044, close to the observed null dispersions of 0.046 and 0.048, cover 0.950, 0.935, 0.975, and 0.960. This limited simulation doesn't validate all departures present in the study.

A second simulation with 2,000 replicates per population and 4,999 draws scores four interval procedures on the same simulated moments. Table 8 lists the populations, each with 125 configurations in 25 recipe clusters, three runs, and ten benchmarks. The wild bootstrap-t covers between 0.928 and 0.954. The cluster $t(24)$ tracks it within 0.02 but falls to 0.911 with one recipe at six times the variance.

The configuration percentile bootstrap falls to 0.872 under recipe-shared run effects. The recipe percentile bootstrap covers between 0.857 and 0.940 with undefined draws in the accuracy-like noise cells. Cross-half error correlation of 0.1 biases the estimate by -0.014 and -0.016. Its margin-like

wild interval misses above in 0.066 of replicates against a nominal 0.025, which puts the shortfall on one side. The accuracy-like population with mean correlation 0.018 carries a bias of -0.007.

We later added a centred cluster test-inversion interval, whose accepted set can be unbounded and is bounded in 23,999 of the same 24,000 simulated datasets. It covers between 0.932 and 0.957 and holds 0.955 with one recipe at six times the variance, where the wild interval gives 0.930. A restricted variant without centring covers between 0.946 and 0.970, above 0.95 in eleven of the twelve populations. A repeat of all twelve populations at 10,000 replicates puts the centred inversion between 0.935 and 0.956 and the wild interval between 0.933 and 0.953.

The Monte Carlo standard error of 0.005 is the size of the shortfalls in Table 8. We repeated every cell at 10,000 replicates with new seeds. No cell moves by more than 0.011 under that repeat. In the four cells that matter most, the wild interval covers 0.933 with cross-half correlation 0.1, 0.935 with one dominant recipe, 0.940 in the margin-like cell, and 0.945 in the accuracy-like cell.

A third seed moves the margin-like cell to 0.951 before it settles at 0.946 over 60,000 pooled replicates, while the other three repeat within 0.004. The centred inversion covers 0.935, 0.956, 0.942, and 0.947 in the same four cells. It keeps its advantage of 0.021 under one dominant recipe while matching the wild interval elsewhere. A further sweep with a third seed puts the wild interval between 0.929 and 0.950 across all twelve. The shortfall belongs to the construction and doesn't come from one set of draws.

We then applied all eight constructions to the data itself with 9,999 draws over twenty bootstrap seeds. On the full battery the margin interval runs from 1.141 to 1.339 under Rademacher weights, 1.130 to 1.349 under residual scaling, and 1.095 to 1.323 under the centred inversion. All eight exclude one on margins and include one on accuracy. Removal of BoolQ leaves both scales above one under all eight, where the widest margin interval is the restricted inversion at 1.391 to 1.840. A separate simulation scored an independent population at one in ten, one in twenty and one in a hundred, and it gave 0.896, 0.951 and 0.990, so the interval tracks the level it claims. A band-shared quarter breaks all three levels together, at 0.743, 0.828 and 0.928.

We compared six further constructions on the same twelve populations. A wild bootstrap-t whose recipe residual is divided by $1 - h_g$, with h_g the recipe's share of the denominator sum, covers between 0.943 and 0.958 at 4,000 replicates. Our version of the interval covers 0.932 to 0.951 there. At 10,000 replicates it improves every cell by between 0.007 and 0.014, reaches 0.95 in ten cells where ours reaches it in one, and runs 4 to 13 percent wider at the median.

We picked the $1 - h_g$ construction after we saw the first four cells. The eight cells that followed confirm it on the same generator family without testing it independently, and we report our interval throughout. On the observed data it gives $[1.130,\ 1.349]$ for margins and $[0.987,\ 1.163]$ for accuracy. That moves no conclusion, although it erases the 0.011 of separation from the planning document's threshold of 1.349 that we report below.

Prepivoting measures calibration inside the resampling model that the interval already assumes. A violation the bootstrap shares with the estimator stays invisible to it. Beran prepivoting, a second wild bootstrap of 1,999 draws inside each of 1,999 outer resamples, calibrates the full-battery margin interval to $[1.151,\ 1.328]$. That interval runs 0.018 narrower than our $[1.143,\ 1.338]$ and still excludes one. Accuracy calibrates to $[0.994,\ 1.161]$ and still includes one, while the BoolQ-removed margin interval calibrates to $[1.689,\ 1.878]$. In the four simulated cells that carry the shortfall it gives 0.924, 0.933, 0.931 and 0.944 against 0.938, 0.931, 0.936 and 0.953 for our interval, so it repairs none of the shortfall.

The estimator assumes that item noise in one half doesn't correlate with item noise in the other. We swept the cross-half noise correlation from -0.1 to 0.5 with 1,000 replicates at each value. The mean margin estimate stays between 0.997 and 1.001 and no margin replicate reaches the observed 1.244, because such noise pulls a real inflation toward one. On accuracy the null dispersion is wide enough that 0.19 of replicates reach the observed 1.078 with no correlation at all.

A scalar effect shared across benchmarks and repeated across halves is a different matter, because the numerator absorbs all of it and the denominator only a tenth. We first swept its share of item-noise variance in a simulated population with no seed covariance, with 2,000 replicates at each value. In that population, with item noise of 0.73 on a unit-variance latent, a share near 0.124 reproduces the observed 1.244, and at a share of 0.05 only 0.007 of replicates reach it, while the accuracy

population, with item noise of 1.45, reaches the observed 1.078 near a share of 0.009. These shares belong to the simulated noise scales and don't describe the released scores.

We then injected the component into the released per-half item noise of each benchmark, as one run-level scalar that all ten benchmarks share and that is identical in both halves, and we scaled each benchmark's component to its measured per-half item noise, the within-half variance less the cross-half covariance, which by construction excludes any component identical in both halves. On margins a share of 0.124 of that measured noise with no seed covariance lifts inflation only to 1.024, with a cross-format diagnostic of 1.010. A share with the whole observed excess would be 1.422 times that measured noise, a value the definition leaves unconstrained, but it would push the cross-format diagnostic to 1.113, which the observed interval $[0.801,\ 1.035]$ excludes. Under zero cross-format run covariance the largest share that keeps the cross-format diagnostic within the observed upper limit of 1.035 is 0.420, and it carries 0.319 of the margin excess on the inflation scale and 0.295 on the covariance scale, and no positive share fits the point estimate, since a value below one already requires negative cross-format seed covariance, which could also offset a larger component. On accuracy the limit allows a share of 0.072, above the 0.053 that explains all of 1.078, so it doesn't bind.

The addition of a 0.124 share to the observed products instead gives a margin cross-format value of 0.933 $[0.808,\ 1.050]$. On accuracy a share of 0.124 alone gives 1.177 and a cross-format value of 1.107, which the accuracy interval $[0.848,\ 1.063]$ excludes, while the share of measured per-half item noise that explains the observed 1.078 is 0.053 and implies 1.047, inside that interval. A uniform component shared by every benchmark is therefore excluded on margins as the whole explanation only under zero cross-format run covariance, at 1.113 under noise-proportional scaling. The observed 0.921 also falls below the 1.153 reference for uniformly spread covariance (Appendix C). A component that follows scoring format, however, stays unmeasured in DataDecide. A regression diagnostic can't separate such a component from a seed covariance that follows the item-noise scales, so that check has no power.

A permutation of run deviations across the items of one group, separately for every run, breaks only the tie between particular items and particular runs. Over 2,000 permutations of the 37,682 items in their 66 groups, 0.087 of the permutations reach or exceed the observed margin 1.244, and 0.352 reach or exceed the observed accuracy. The item-level channel therefore accounts for about 0.001 of the margin estimate and 0.002 of the accuracy estimate, against excesses over one of 0.244 and 0.078. When we redraw the item half split 200 times in that kernel, separately from the fifty re-splits of the primary run, the margin range is 1.242 to 1.254 and an accuracy median of 1.083 with a range of 1.059 to 1.103.

The shipped 1.244 and 1.078 are therefore not a favourable draw. The real re-split standard deviations of 0.0018 and 0.0083 exceed the permuted 0.0009 and 0.0053. Item-level structure does survive, and it widens the spread but leaves the centre in place. An effect that every item of a benchmark carries survives this permutation untouched. Only the sub-benchmark form of the violation is bounded here.

Two scenarios fitted to these data, each with 2,000 replicates and 125 configurations, draw the seed covariance from the observed estimate. The second also scales each recipe's seed deviations by its observed denominator share. In the first, margin coverage is 0.954 for the wild interval, 0.946 for the cluster-robust t, 0.954 for the centred inversion, and 0.970 for the restricted inversion. In the second, the wild interval holds 0.946 and 0.945, the cluster-robust t falls to 0.901 and 0.927, while the centred inversion turns conservative at 0.977 and 0.958. The $G/(G-1)$ correction to the cluster-robust t reaches only 0.939 and 0.909 on margins in the two scenarios. These scenarios aren't independent confirmation, since they are fitted to the same data that the intervals describe.

On the observed data the centred inversion gives 1.095 to 1.323 for margins and 0.995 to 1.162 for accuracy, beside wild intervals of 1.143 to 1.338 and 0.993 to 1.157. The upper endpoints differ by at most 0.015, but the inversion puts the margin lower endpoint 0.048 lower. The wild lower endpoint shouldn't be read as a firm floor. In the size bands the inversion gives 0.765 to 1.737 for 530M margins, which includes one although the wild interval excludes it, and 1.024 to 1.398 for 530M accuracy, which still excludes one. The accuracy intervals after BoolQ and WinoGrande removal stay above one at 1.397 to 1.716 and 1.026 to 1.184.

Table 8. Coverage of nominal 95 percent intervals across 2,000 simulated populations per cell. Bias is the mean estimate minus the population Λ. Noise gives the item-noise standard deviation relative to unit run effects. Inversion is the centred cluster test-inversion interval, computed later on the same simulated moments. Every cell was afterwards repeated at 10,000 replicates, and the inversion range in the main text comes from that repeat, so it differs from this column by a few thousandths.

Population	Λ	Bias	Wild	Cluster t	Config.	Recipe	Inversion
Null, noise 0.73	1.000	0.000	0.953	0.946	0.948	0.937	0.953
Null, noise 1.45	1.000	−0.011	0.943	0.941	0.943	0.857	0.946
$\bar{r}_E = 0.061$, noise 0.73	1.245	0.001	0.947	0.943	0.938	0.929	0.948
$\bar{r}_E = 0.018$, noise 1.45	1.078	−0.007	0.943	0.940	0.942	0.910	0.945
One trait with 0.68 of trace	1.152	−0.001	0.938	0.936	0.937	0.925	0.941
One trait with 0.84 of trace	1.031	0.000	0.954	0.950	0.955	0.940	0.957
Recipe-shared run effects 0.5	1.245	−0.003	0.944	0.940	0.872	0.931	0.947
Student t, 4 df	1.245	0.000	0.946	0.930	0.943	0.927	0.955
One recipe at 6 times variance	1.245	−0.001	0.930	0.911	0.931	0.912	0.955
Cross-half correlation 0.1, noise 0.73	1.245	−0.014	0.928	0.926	0.930	0.918	0.932
Cross-half correlation 0.1, noise 1.45	1.078	−0.016	0.946	0.941	0.943	0.933	0.948
$\bar{r}_E = 0.2$	1.673	−0.002	0.947	0.943	0.943	0.928	0.948

The analysis also reports 1,000 trait-specific run-label permutations, each using 4,999 bootstrap draws, whose intervals contain the reference value one in 0.925 of margin permutations and 0.954 of accuracy permutations. Our repeat with 2,000 permutations gives 0.944 on margins and 0.949 on accuracy. Ten further runs at 1,000 permutations put margin coverage between 0.915 and 0.942, and that record therefore reads as one coarse draw. The margin interval falls below the null in 0.032 to 0.052 of permutations against a nominal 0.025. That lower-tail imbalance matters for an upper bound below the planning document's margin threshold of 1.349 (Appendix B), where the reported upper endpoint leaves only 0.011 of separation.

These are conditional randomisation results, contingent on the exchangeability assumed by the permutations, and they aren't coverage measurements for the unknown nonzero covariance of the original population. The cluster-t comparator has recorded coverage of 0.881 and 0.941 against our repeat's 0.901 and 0.938. The accuracy upper-tail fraction is 0.021 in the record and 0.020 in the repeat, close to the 0.028 of the earlier 500 draws.

The corrected accuracy interval includes one, since the reported delete-one-recipe jackknife standard errors of 0.049 and 0.039 give approximate $t(24)$ intervals of 1.143 to 1.345 for margins and 0.998 to 1.158 for accuracy. Across the 1,000 permutations the estimator standard deviations are 0.0488 for margins and 0.0382 for accuracy, against N1 values of 0.046 and 0.048. The accuracy discrepancy therefore concerns simulated versus conditional permutation populations and points to no error in the observed bootstrap scale.

C.16 Availability of a separate competence measurement The desired gain control requires per-run loss on held-out text separate from the benchmark items. The DataDecide fields inspected in the source analysis don't supply it. The PolyPythias evaluation files considered there cover three sizes at early and final checkpoints, yet they hold accuracies, likelihood differences, and LAMBADA perplexity on the same benchmark items, and that perplexity doesn't provide the required independent measurement scale.

The Mistral release of Karamcheti et al. (2021) provides five GPT-2 Small and five GPT-2 Medium runs with about 600 checkpoints each, listed as 610 in the release summary, with WikiText-103 and OpenWebText validation perplexity. The source search didn't find the corresponding per-run benchmark battery. Without paired held-out loss and benchmark scores from the same runs, we can't use that release to identify the gain contribution.

C.17 Competence proxy audit and calibration The adjustments regress each benchmark's run contrast on a gain covariate s_{cr}, the mean per-byte answer log-likelihood, and on a leave-one-benchmark-out competence proxy $x_{cr}^{(-j)}$ fitted on the opposite item half, as $y_{crj} = \alpha_{cj} + \beta_j x_{cr}^{(-j)} +$

$\gamma_j s_{cr} + u_{crj}$ with one slope per benchmark over 250 run contrasts, and we treat them as descriptive. We audited the shipped competence adjustment and two repairs on the 375 reduced runs, then applied each adjustment to simulated populations with known seed covariance. The audit reproduces the shipped after-adjustment estimates of 1.244 and 1.076. The variance floor binds for WinoGrande on margins and for PIQA and WinoGrande on accuracy, and WinoGrande then supplies more than 99.99 percent of the squared standardised inputs to the margin proxy. The shipped margin adjustment in effect regresses every other benchmark on WinoGrande. Removal of WinoGrande from the proxy gives 0.890 with interval 0.805 to 0.968, while removal of any other benchmark leaves margin inflation between 1.244 and 1.245.

The pooled in-sample fit over the 250 run contrasts gives margin and accuracy inflation of 1.275 and 1.097 after gain adjustment, 1.276 and 1.094 after gain and competence together, and 1.244 and 1.076 after competence alone. The repair we fixed before we saw its output standardises by seed variances shrunk halfway toward their median positive value, cross-fitted over recipe folds. It gives margin inflation of 0.731 with interval 0.673 to 0.784, and accuracy inflation of 1.004 with interval 0.925 to 1.075.

A split-half check shows that our proxy carries no measurable signal about the run it scores, since its half-A and half-B scores correlate at -0.086 on margins and -0.072 on accuracy. We also fitted four proxy weightings out of fold, and a one-factor proxy with regression-score weights reaches 0.965 and 0.738 on the same check.

Cross-fitted margin inflation is 1.245 under the shipped weighting, 0.731 after shrinkage, 0.973 for the factor proxy, and 0.889 when the clipped benchmarks are dropped. A 999-draw recipe bootstrap that refits the whole pipeline gives 0.935 to 1.018 for the factor proxy on margins and 0.970 to 1.104 on accuracy. The adjusted margin estimate therefore no longer separates from independence. None of the four manufactures covariance where none exists, since under simulated independent seed deviations they all return a mean between 0.998 and 1.003. None of this identifies mediation either, because every one of these proxies averages the same seed deviations that it adjusts and so removes shared covariance by construction. With the observed seed covariance and no competence variable at all, the four weightings return margin means of 1.097, 0.745, 0.980, and 1.029.

The observed cross-fitted values sit inside the central 90 percent of those simulations for the shipped, shrunk, and factor weightings on both scales. None of them carries evidence of a competence mechanism beyond covariance removal. Those simulations refit the cross-fitted pipeline in every replicate. Their percentiles describe a different estimator from the in-sample matched means in Table 9. The clipped-drop weighting is the one departure, since its observed cross-fitted 0.889 on margins and 0.994 on accuracy fall below simulated fifth percentiles of 0.908 and 1.003. Under a common correlation of 0.178 the factor weighting returns 0.852 on margins, so these proxies can overshoot under equicorrelation.

The in-sample adjustments in Table 9 repeat that pattern. With the observed seed covariance the repaired margin adjustments return 0.745 to 0.889, each within 0.015 of the observed 0.731 to 0.890, and on accuracy they lower the matched simulated mean from 1.116 to between 1.049 and 1.079. The observed values of 1.004 to 1.019 lie inside the central 90 percent of those simulations. A competence adjustment therefore needs a measurement taken outside the benchmark scores, such as held-out loss on the same checkpoints.

We then checked whether those checkpoints are reachable. An index built from the model repositories matched 120 of the 375 runs to a training branch whose released steps overlap the run's evaluation steps, with a smallest overlap of 0.042, and each of those 120 runs has a released checkpoint at its selected step. We had built the remaining names from evaluation tables that spell 17 of the 25 recipes differently, and because the Hub answers a missing repository with HTTP 401 instead of 404, we read the refusals as gating. Under the released naming all 125 repositories resolve, and every one of the 375 runs reaches a public commit with config, tokenizer and weights. Among the 120 matched runs the branch's final step disagrees with the run's final evaluation step in 21 cases, a further obstacle to scoring loss at the step the scores describe. The measurement is therefore open to anyone with the compute.

C.18 Transport to PolyPythias The planned transport analysis scores nine runs per configuration on a nested 4,755-item battery, and for three sizes the source pipeline gives margin inflation of

Table 9. Inflation after each competence adjustment, observed and in simulated populations with 300 replicates each. Simulated columns give replicate means under independent seed deviations, a single common correlation whose population inflation equals each scale's observed value, and the nearest positive semidefinite matrix to the observed seed covariance.

	Margin				Accuracy			
Adjustment	Observed	Indep.	Common	Matched	Observed	Indep.	Common	Matched
None	1.244	1.000	1.242	1.243	1.078	0.998	1.093	1.116
Shipped, floored	1.244	1.000	1.170	1.093	1.076	0.999	1.092	1.109
Floored traits dropped	0.890	1.001	0.948	0.889	1.008	0.999	1.083	1.079
Shrunken, in sample	0.732	1.002	0.876	0.747	1.019	0.999	1.061	1.050
Shrunken, cross-fitted	0.731	1.002	0.875	0.745	1.004	0.999	1.060	1.049

1.40574 compared with 1.24395 for DataDecide, an excess ratio of 1.66323. On excess inflation the DataDecide prediction falls approximately 40 percent below PolyPythias, while the raw inflation difference is approximately twelve percent and falls inside the proposed twenty-percent band. The planning document names aggregate standard deviation but doesn't resolve this scale choice, which makes the excess-based failure assessment post hoc.

An exploratory extension adds the 14M and 31M configurations with nine runs each at step 143000. Within that rescore margin inflation changes from 1.41372 at three sizes to 1.26199 at five sizes, while the excess ratio moves from 1.69592 to 1.07393. Accuracy changes from 2.28589 to 1.71085 against DataDecide's 1.07837, and only the three-size accuracy point estimate exceeds the stated detection threshold of approximately 2.05. The configuration-bootstrap interval for five-size accuracy runs from 0.869 to 2.428, and at three sizes its lower endpoint is undefined, with an upper endpoint of 2.971.

The margin intervals are 0.939 to 1.781 at three sizes and 0.927 to 1.537 at five sizes, and both contain one and DataDecide's 1.24395. Since the bootstrap resamples only three or five configurations, we also inverted the centred cluster test described above with each configuration as its own cluster, and that gives margin sets of 0 to 2.106 at three sizes and 0.519 to 1.664 at five sizes. Since it can't exclude any nonnegative accuracy value, these data don't bound PolyPythias accuracy inflation.

A repeat on two T4 GPUs with the same scorer, token cap, items, and checkpoints reproduces every reported estimate, interval, and standard error. The rescore itself used one T4 with a 6,000-token cap instead of 30,000 tokens, and its three-size margin value of 1.41372 differs from the pipeline's 1.40574 by 0.6 percent. The repeat gives the five-size excess ratio of 1.07393 stated above and adds the inversion sets.

Against a single configuration's ratio the decomposition doesn't show out-of-sample value. Nine seeds per configuration let us ask whether a three-run fit predicts the measured ratio, which is the standard deviation of the battery mean divided by the root of the weighted sum of per-benchmark variances, item noise included. For each of the 84 ways to choose three runs, we predicted it on the six held-out runs, where independence predicts one, the plug-in predicts the inflation estimate, and the decomposition adds each benchmark's estimated item noise to both sides. On margins across three sizes the mean absolute log error is 0.339 for independence, 0.536 for the plug-in, and 0.512 for the decomposition, so neither fitted prediction beats the constant.

On accuracy the same three errors are 0.370, 0.447, and 0.384. The decomposition, however, beats independence in 0.50 of splits. The six-run target is itself noisy, and the measured margin ratio runs from 0.70 to 1.85 between its fifth and ninety-fifth percentiles.

A target pooled across configurations changes that answer and shows how much of the first comparison was target noise. With one held-out ratio built from mean within-configuration variances, the decomposition has log error 0.275 against 0.319 for independence on three-size margins and 0.367 against 0.523 on accuracy. It beats the plug-in on every one of these sets. The covariance term, however, still hurts on five-size margins. The 84 splits share runs, and these differences are therefore descriptive, without an interval.

Table 16 of the extended supplement gives fold-mean point errors, and Appendix C.8 gives paired intervals for their differences. P2 uses performance correlation from the held-out fold itself because its calculation needs no seed replication. Negative held-out variance estimates are clipped to zero before model selection, and across the five folds that clipping touches two margin traits and five accuracy traits. Because shared clipping doesn't guarantee unchanged rankings, we repeated the comparison under two alternative rules with the same 2,000-draw paired bootstrap as Appendix C.8.

On margins, P1 and P3 stay close under both rules, at 0.0063 each under the drop rule and at 0.0039 and 0.0045 under training fill, while independence stays near 0.035 under both alternative rules. The P1G and P3 intervals for a difference from independence still exclude zero under both rules, though under training fill the P1 interval runs from -0.059 to 7×10^{-5} and so includes zero. On accuracy, the drop rule leaves P3 the best model at 0.0090 and the fill rule gives P1 and P3 errors of 0.0122 and 0.0118, and none of the accuracy intervals excludes zero under either rule.

We simulated both PolyPythias rules of Section 4.3 after we fixed them, with nine seeds at each of five sizes, seed effects drawn from the DataDecide margin covariance clipped to positive semidefinite, item noise from its margin reliabilities with WinoGrande's set to 0.5, and 4,000 replicates. The first rule, computed with the committed delete-one-seed jackknife that recentres the remaining seeds, passes in 0.259 of replicates at a true 1.244 and in 0.016 at independence. The middle 90% of estimates runs from 1.026 to 1.456. A jackknife that drops a seed's term but doesn't recentre the others understates the standard error and on the same draws passes in 0.356. The uncentred script also assigns the two ARC reliabilities to each other's benchmarks, and we kept its choice so the figures stay comparable.

Because the item noise comes from DataDecide, we reran the first rule with margin reliabilities estimated on the 45 PolyPythias transfer runs, pooled over sizes or separately by size, both at their 4,755 items and scaled to bank one's item counts, with any trait whose seed or noise variance estimate isn't positive set to 0.5. On the rule-two simulation's draws, where the DataDecide noise passes in 0.257 at a true 1.244 and in 0.014 at independence, the four PolyPythias versions pass in 0.205 to 0.257 at a true 1.244 and in at most 0.015 at independence. The widest middle 90% of estimates at a true 1.244, from 0.993 to 1.488, comes from the per-size reliabilities at 4,755 items. The two versions scaled to bank one, which the rule actually scores, pass in 0.254 and 0.257.

For the second rule the simulation adds a bank two with one sixth of bank one's items, close to the built ratio of 6,808 to 37,682. At a true 1.244 with no shared item component the rule returns not supported in 0.226 of replicates, supported in 0.013 and undecided in 0.761. Under independence with a uniform shared item share of 0.124, it returns supported in 0.033 and undecided in 0.957, and on this family's item noise that share lifts the median within-bank estimate only to 1.046 against a cross-bank median of 0.999. That share falls far short of the inflation it would need to explain.

Under independence a uniform share of 0.755 makes the expected within-bank inflation 1.244 on this noise, and at that share the first rule passes in 0.338 of replicates while the second returns supported in 0.067 and undecided in 0.919, with median within-bank and cross-bank estimates of 1.240 and 1.003. On PolyPythias item noise scaled to bank one the matching share is 0.848. The second rule returns supported in 0.070 and undecided in 0.916. At the 4,755 transfer items the share is 0.230, with supported in 0.026 and undecided in 0.958. The second rule therefore rarely detects a shared component large enough to explain the inflation, returns not supported in at most 0.017 of replicates at that share, and an undecided verdict can't exclude one.

C.19 External random-seed panel This panel supplies single-unit estimates without cross-half correction or uncertainty intervals, and that limits comparison with the corrected primary estimator. The signal-and-noise runs don't share models with the DataDecide analysis, although the releases come from one laboratory, use a shared scoring harness, and overlap in all ten study traits. The check averages estimates over twenty released checkpoints and uses ten initialisation runs or nine data-order runs. Table 10 gives those estimates across the twenty checkpoints.

The bits-per-byte likelihood estimates exceed one under both seed mechanisms. The unadjusted accuracy values, however, lie near or below one. Adjustment increases three of the four estimates and reduces only the initialisation estimate on that scale, from 1.705 to 1.464, a pattern that doesn't support a stable interpretation of the adjustment at one unit. Independent item noise would attenuate covariance ratios toward one under the relevant assumptions, though this arm doesn't measure the

Table 10. Inflation estimates in the external random-seed panel across twenty checkpoints. The leave-one-out adjustment controls each trait deviation with the mean deviation of the other traits.

Seed variation	Score scale	K	Λ	Adjusted Λ	$\bar{r}_E$
Initialisation	bits per byte	18	1.705	1.464	0.112
Initialisation	accuracy per character	8	0.989	1.105	-0.003
Data order	bits per byte	18	1.625	1.763	0.096
Data order	accuracy per character	8	0.868	1.059	-0.035

amount of attenuation or verify those assumptions. We therefore use this panel to compare score scales, with the size of item-noise attenuation still unmeasured.

C.20 Checkpoint noise as a proxy for seed noise Heineman et al. (2025) measure a benchmark's noise as the spread of its score over the final checkpoints of one run and compare that spread with seed variation one benchmark at a time. For an average the two measures agree only when both noise sources share one pattern of correlation across benchmarks. We compute the inflation of the equal-weight average under each from their released evaluations. Checkpoint inflation pools the covariance of each run's final five checkpoints around the run mean, and seed inflation uses the covariance across runs at the final step. Intervals come from 5,000 bootstrap draws of runs, or of recipes on DataDecide.

On the 225 DataDecide runs with five released checkpoints, checkpoint inflation of the ten-benchmark accuracy average is 1.104 $[1.057, 1.156]$, which lies inside our seed-based interval around 1.078. Per-byte margins give 1.574 $[1.058, 1.624]$, with the point estimate near its own upper limit and per-size values from 1.103 to 1.832, a sign that a few runs dominate the pooled covariance, so we don't compare margins.

The OLMo 2 1B runs of the same release carry ten initialisation seeds and eight of our benchmarks, without BoolQ and OpenBookQA. There, checkpoint inflation of accuracy per character is 0.992 $[0.890, 1.087]$. Seed inflation at the final step is 0.516 $[0.218, 0.765]$, but that step gives the lowest of the last 30 checkpoints, whose single-step seed values run from 0.516 to 1.301 with a median of 0.986. The run bootstrap at one step misses that spread.

We read the median, which agrees with the checkpoint value and with the twenty-checkpoint mean of Table 10. Bits per byte give checkpoint inflation of 1.112 $[0.995, 1.207]$ against a median seed value of 1.014, and nine data-order runs give medians of 0.910 on accuracy and 1.060 on bits per byte. On both releases checkpoint noise thus tracks the seed inflation of an accuracy average, and a seed estimate from ten runs at one checkpoint can land anywhere in that range. These comparisons are exploratory, rest on ten runs of one configuration, and omit BoolQ, which carries most of our margin covariance.

C.21 Gain calibration and recipe split sensitivity Table 17 of the extended supplement compares the default gain simulator, which applies a shared margin scale across traits, with the matched simulator, which copies each benchmark's margin and variance components from the data. At $\sigma_g = 0.05$ the two reported excess shares differ by a factor of about 7.5. The subtraction of estimated leakage variance 1.91×10^{-3} from total gain variance 3.67×10^{-3} gives a gain standard deviation of approximately 0.042, and it depends on the variance decomposition assumed for the covariate.

The G5 specification fails its threshold at the default 0.05 cell, which matches Table 9 of the extended supplement within Monte Carlo error. Accuracy estimates stay within 0.004 of one, and that doesn't establish absence of a gain mechanism since positive multiplication preserves signs. A recipe resample puts a 95 percent percentile interval from 0.011 to 0.063 on the gain standard deviation, and at those endpoints the default simulator gives margin inflation of 1.006 and 1.221, which are shares of 0.02 and 0.91 of the observed excess. The observed seed variances lie below 0.002, while the matched simulator inflates accuracy seed variances by factors of 2.4 to 116. Applied to the default and the matched simulated populations, the gain estimator recovers only 0.037 and 0.036 against an input of 0.042.

The reconstructed split uses the plan's master seed, though the plan didn't specify a complete reconstruction rule and no contemporaneous split artifact was recovered. For a recipe subset S, the estimator is $\widehat{\Lambda}_S = \sqrt{\sum_{c \in S} T_c / \sum_{c \in S} U_c}$. The analysis enumerates all $\binom{25}{17} = 1{,}081{,}575$ seventeen-recipe estimation sets. Margin inflation runs from 1.037 to 1.317 across them and accuracy from 0.963 to 1.191. No set reaches the plan's thresholds of 1.349 and 1.40. Table 18 of the extended supplement summarises that exhaustive sweep over every admissible estimation subset.

No subset reaches either proposed point threshold on either scale. Because the sweep runs after the analysis choices were made, it provides no check on choices that used the full dataset.

We also recomputed our interval after the deletion of each recipe, recipe pair, size band and benchmark in turn. The margin interval excludes one under every recipe, pair and benchmark deletion. The one exception is the 530M band, whose removal leaves 1.178 with an interval of 0.993 to 1.338, a lower endpoint 0.007 below one. Simulation puts at least one band-deletion failure in 0.164 of populations whose full interval excludes one.

One failure out of five is close to chance. At a true ratio of 1.25 under compound-symmetric correlation and margin item noise of 0.73, 125 configurations and three runs clear one in 0.970 of replicates, while a simulated accuracy-like population at a true ratio of 1.078 clears it in 0.111. Re-estimation on each remaining pair of runs leaves the margin estimate at 1.256, 1.242 and 1.235, and all three exclude one, while the accuracy estimate sits at 1.084, 1.114 and 1.030, and none of its three intervals excludes one.

The accuracy conclusion is fragile in the opposite direction. Its full-battery interval includes one yet excludes it under five of the 25 recipe deletions, under removal of the 750M band, and under removal of WinoGrande or BoolQ. All eight exceptions put the whole interval above one, with the BoolQ removal's lower endpoint at 1.412. Among simulated null populations with a true ratio of one whose full interval includes one, five or more recipe deletions flip in 0.018, and a second null calibrated to the observed 1.078 puts that at 0.032. The observed five flips therefore sit in the upper few percent of what a deletion sweep produces by chance, and all move towards more inflation.

With clusters on the five size bands instead of the 25 recipes, the margin interval runs from 1.030 to 1.426 and still excludes one. The matching accuracy interval, however, runs from 0.961 to 1.184 and still includes one. The two-way clustered variance turns negative on full-battery margins and returns an implausibly small accuracy standard error. We report neither, and five size clusters are too few for it.

Without BoolQ it behaves and leaves both scales above one, the accuracy one at 1.276 to 1.796. A two-way wild bootstrap avoids the negative variance, but at 8,000 replicates its coverage sits 0.008 to 0.010 below the recipe-clustered interval, more than four standard errors. We keep the recipe clustering for that reason. A first comparison over 1,000 replicates at 999 draws had left that difference inside Monte Carlo noise.

A rescoring of 150 ARC-Easy documents matches each branch to its released seed in all 24 cells at 150M and all fifteen loaded at 1B. We ran that check because the release shares one seed label across every recipe in a band. Seventeen recipes went unchecked because of the repository misnaming described in the competence proxy audit above.

We simulated a run effect that follows the size band instead of the recipe, with the true ratio fixed, and pooled two runs to 12,000 replicates per share. At a share of zero the recipe-clustered wild interval covers 0.943 and the size-clustered one covers 0.951. At a quarter the recipe-clustered interval falls to 0.821 and at a half it reaches 0.580, while the size-clustered cluster-robust t interval holds 0.935 and 0.898. A recipe-shared effect swept over 0, 0.114, 0.25 and 0.50 leaves the cluster-robust t interval at 0.947, 0.944, 0.929 and 0.869. On accuracy the band sweep leaves the recipe-clustered interval at 0.950, 0.947, 0.943, 0.916 and 0.801 across shares of 0, 0.044, 0.114, 0.25 and 0.50, while the size-clustered wild interval covers only 0.832 even without sharing, so that fallback needs its t version.

The recipe-clustered interval stays between 0.942 and 0.952 as runs per configuration vary from two to ten. Item noise swept from 0.25 to 3.00 leaves coverage between 0.944 and 0.954 through 2.00 and collapses it to 0.668 at 3.00, while the exclusion rate at a true 1.2446 falls from 0.998 at the quietest cell to 0.326 at 2.00. Missing cells and missing seeds cost width and power but leave coverage intact. A two-fold or four-fold spread of item noise across size bands on margins leaves

coverage between 0.919 and 0.945. The loudest band breaks the estimator and the unevenness itself doesn't. A four-fold ramp at the accuracy battery's base noise takes coverage to 0.208 with 0.776 of replicates undefined, and a single loud band holds 0.947 on margins while falling to 0.291 on accuracy.

A sixteen-fold ramp on margins takes coverage to 0.111 with 0.885 of replicates undefined. Among small designs only 25 recipes with four runs reach an exclusion rate of 0.997. On accuracy the cheapest corner fails, since ten recipes with two runs cover 0.602 with 0.369 of replicates undefined. At five recipes the wild interval covers 0.749 on margins, below the 0.910 of the plain cluster-robust interval, and 0.360 on accuracy with 0.606 of replicates undefined. A correlation of 0.02 or 0.05 between the two item halves leaves coverage at 0.946 and 0.943, 0.10 gives 0.928, and 0.40 gives 0.864. Our passage-level check moved the estimates by less than 0.001.

At the accuracy battery's noise the exclusion rate runs 0.018 at a true ratio of exactly one, 0.063 at 1.05, 0.135 at 1.10, 0.281 at 1.15, 0.668 at 1.25 and 0.968 at 1.40. At our 125 configurations and three runs, a simulated accuracy-like population at a true ratio of 1.078 clears one in 0.111 of replicates, and a recipe count of 50, 100 and 200 at three runs takes that to 0.165, 0.306 and 0.620. Six runs per configuration take the same four designs to 0.196, 0.398, 0.704 and 0.947. Four fifths needs between 500 and 1,000 configurations at six runs, eight to sixteen times the run budget of this release.

At the margin battery's noise the rate reaches 0.652 at 1.15 and 0.970 at 1.25. Coverage at that noise is 0.944 when the truth is exactly one, then 0.934, 0.939, 0.949, 0.946 and 0.943 at 1.05, 1.10, 1.2446, 1.40 and 1.60. There the interval excludes one in 0.018 of replicates at a true ratio of one, the false positive rate to assume with no dependence. Five explicit covariance shapes cover between 0.936 and 0.946, and a gap between two of them vanished on a 12,000-replicate repeat. Two shapes with true ratios of 0.9927 and 0.9900 exclude one in 0.011 and 0.009 of replicates.

We then measured the band-shared component directly with a batch-slot permutation null, whose bound is anti-conservative where it is meant to exclude. Over 2,000 permutations the observed statistic sits inside the null on all four batteries. The share comes to 0.008 on full-battery margins with a 95 percent upper bound of 0.044. A follow-up sweep at shares of 0.02, 0.05 and 0.10 with 4,000 replicates each gives recipe-clustered coverage of 0.946, 0.941 and 0.929.

Coverage stays near nominal up to twice the bound. On simulated populations at 500 replicates and 999 permutations the test rejects 0.044 of the time at a true share of zero, and its power reaches only 0.490 at the 0.044 bound. On the accuracy batteries it rejects 0.048 at zero and its power falls to 0.308 at the bound. The quoted standard error understates the actual spread at large true shares. The mirror test over the five sizes of a recipe carries a standard error of 0.067 on the share against 0.018 for the band version, and it bounds recipe sharing only below a share of 0.114.

We also simulated the corner where both channels sit at their bounds. Recipe sharing alone at 0.114 leaves the recipe-clustered interval at 0.949, and twice both bounds takes it to 0.927. A first pass separated band sharing alone from the joint corner by 0.012, but a 12,000-replicate repeat puts them at 0.9444 and 0.9428 with median widths of 0.242 and 0.246, and that separation was therefore noise. The CR3 correction lifts the interval from 0.942 to 0.950 at the band corner and from 0.946 to 0.954 at the joint corner, and we report the uncorrected interval, which is what the shipped estimator computes. The same five cells at the accuracy battery's noise cover 0.950, 0.942, 0.951, 0.946 and 0.940, while on that scale the correction also moves the no-sharing cell from 0.949 to 0.956.

On the size-clustered fallback it takes coverage from 0.841 to 0.751 and raises the undefined share from 0.119 to 0.229. The sharing our clustering assumes is a precaution these data can neither confirm nor exclude, while the sharing that would break it is bounded where it costs under a point of coverage.

C.22 Pre-specified held-out test The protocol fixed the task list, the replacement rule, the statistic, the interval, the pass rule, and a scope rule before any production score existed, and it records every later change as a dated amendment. The OLMES harness at a pinned commit builds every prompt and continuation, MedMCQA contributes 1,000 items drawn under a fixed seed, and DROP and CoQA are each capped at 1,000 items. Halves come from the clustered split of the passage-aware check with a seed fixed in the protocol. Items that share a passage or a story never straddle halves.

The CoQA story grouping gives 479 groups for its 1,000 items. Before production, the six pilot checkpoints of one recipe at 150M and 1B each matched their own release seed on 300 ARC-Easy items. The backup rule then read the pilot's three 1B runs, where the AGIEval task LogiQA-en (Liu et al., 2020) gave mean accuracy 0.269 against a bar of 0.270 and the AGIEval task LSAT-LR (Wang et al., 2022) gave 0.219 against 0.220, with chance at 0.25 and 0.20. Both fell short, so the rule replaced them in its listed order with the Gen2MC multiple-choice versions of DROP and CoQA (Team Olmo et al., 2025b), whose distractors an LLM generated, and it reads accuracy only. Each miss is smaller than the pilot's binomial standard error of about 0.010 over three runs and 651 or 510 items.

The rule therefore decided the swap, and the evidence didn't, and we applied it as written. Before each run was scored, a spread of its items was checked against a one-sequence reference path to within 0.001 nats, with a slower path that agrees as the fallback. Pilot timing projected 66.0 session hours for all 375 runs and 54.9 for the 225 runs in scope, and that estimate triggered the scope rule. That budget also led us to withdraw a six-task secondary analysis that we had pre-specified earlier.

The request file changed twice after the freeze, and neither change touched a prompt, continuation, label, or item order. The first replaced a salted hash in the passage-group labels with a deterministic one. The second regrouped CoQA by story once it became a primary task, and this rewrote the group field on 932 rows. Two production shards were scored on the file before the CoQA regrouping. The analysis reads group labels only from the final file, since scores are keyed by task, document, and choice.

All 225 runs in scope were scored, 205 of them on Kaggle T4 cards and the last 20 at 1B on three RTX 3090 cards with torch 2.14.0 in place of the T4 image's 2.10.0 after the weekly quota ran out, with the same scorer, float32 weights, 6,000-token cap, and pinned transformers 4.57.1, and every one of those 20 runs passed the 0.001-nat batch check below 0.0001. The pilot's three c4 runs at 1B were scored before either change to the file. The test in Section 4.2 reads all three sizes, and three earlier results preceded it. An earlier interim on the 530M runs alone gave held-out margin inflation of 1.198 with interval $[0.864,\ 1.454]$ and accuracy inflation of 1.373 with interval $[1.138,\ 1.567]$, which failed the rule as well. The two finished sizes then gave held-out margin inflation of 1.201 with interval $[0.546,\ 1.616]$ and accuracy inflation of 1.327 with interval $[1.153,\ 1.484]$, which failed the rule at a scope where the original ten benchmarks also failed it, at 1.316 with interval $[0.880,\ 1.637]$.

A second interim, logged in the protocol before either build ran, adds the 18 recipes that the finished shards had scored at 1B to the two finished sizes. On those 68 configurations, held-out margin inflation is 1.181 with interval $[0.685,\ 1.552]$ and accuracy inflation is 1.262 with interval $[1.084,\ 1.422]$, which fails the rule there too, while the original ten benchmarks on the same configurations give 1.174 with interval $[0.954,\ 1.360]$. At the 18 scored 1B configurations alone, the held-out estimate is 1.152 with interval $[0.599,\ 1.516]$. The original ten give 0.947 with interval $[0.295,\ 1.306]$.

The seven recipes missing at 1B sit in the last shard, which means shard order fixed the interim's recipe set and no 1B number did. The analysis kernel first recomputes the shipped margin estimate from the 375 released runs and stops unless it returns 1.24395 to within 0.0005, and it did. The 60 four-benchmark subsets at 530M and 750M come from the battery-size kernel run at that scope before any 750M held-out score existed, and 30 of them fell below 1.201. The all-size subsets in Table 2 come from the curve the protocol committed to, and 32 of their 60 fall below the full-scope 1.217. A three-size curve run after that result, and so post hoc, gives a four-benchmark median of 1.172 with 5th and 95th percentiles of 0.979 and 1.485, with 32 of its 60 subsets below 1.217, the same count as at all sizes, and 0.55 of them exclude one against 0.57 there.

Table 20 of the extended supplement gives the estimate with one held-out task removed at a time. Without CoQA, held-out margin inflation falls to 1.090 with interval $[0.990,\ 1.181]$, and without any other task it stays between 1.120 and 1.213. When the four held-out tasks are pooled with the original ten, margin inflation is 1.419 with interval $[1.051,\ 1.714]$ and accuracy inflation is 1.240 with interval $[0.986,\ 1.458]$. Table 11 gives the battery-size curve at 530M, 750M, and 1B, at 530M and 750M, and at 530M alone alongside the all-size curve that the protocol committed us to report. None of the entries uses a held-out score, since every entry reads subsets of the original ten benchmarks on the released runs.

The held-out tasks carry 1,000 items each. The original benchmarks range from 500 to 14,042 items. A separate kernel asked what item count costs the interval. For each cap it drew that many items per

benchmark under a fixed seed, split them into halves by passage group, and recomputed every run's trait scores on the two halves. The uncapped entry therefore uses the kernel's own split in place of the primary one.

A cap of 1,000 items per benchmark leaves 9,500 of the 37,682 items and gives full-battery margin inflation of 1.218 with interval $[1.121,\ 1.309]$, against 1.247 with interval $[1.146,\ 1.342]$ uncapped. A cap of 200 gives 1.179 with interval $[1.002,\ 1.337]$. Over 20 sampled four-benchmark subsets, the margin share whose interval excludes one stays between 0.45 and 0.55 at every cap from 200 items to the full counts. Item count therefore doesn't explain the four-task failure rate. At the pre-specified scope of 530M, 750M, and 1B, the original ten benchmarks give margin inflation of 1.240 with interval $[1.054,\ 1.397]$ and accuracy inflation of 1.097 with interval $[0.922,\ 1.246]$, both computed from the released original-benchmark scores inside the kernel that reported the test.

Table 19 of the extended supplement gives the rule's simulated pass rate at each scope it could take. The item noise is 0.73 at the margin-like level, 1.45 at the accuracy-like level, and 1.00 between them, and we fixed that design before the two-size analysis ran. At a true ratio of one the pass rate stays between 0.011 and 0.022 at the three fixed noise levels, a sign that the rule holds its nominal size. At two sizes and a true ratio of 1.244, the median simulated interval width is 0.318 at margin-like noise and 0.530 at accuracy-like noise, against the observed 1.070. After we saw that width we added noise levels of 2.00, 2.50, and 3.00, whose two-size median widths at a true 1.244 are 0.765, 0.972, and 1.146.

At 2.50 and 3.00 the pass rate at a true 1.244 is 0.115 and 0.080 against a size of 0.024 and 0.025. The wild interval's coverage there falls to 0.699 and 0.562, and 0.280 and 0.425 of the intervals have an undefined lower limit, the same failure that the tables above show on the data. At three sizes the median widths at a true 1.244 are 0.258, 0.306, and 0.431 at the three fixed noise levels and 0.640, 0.839, and 1.009 at the added ones, against the observed 0.626, and at 2.00 the pass rate is 0.262 with coverage 0.918 and 0.040 of intervals undefined. An exploratory run on the 750M runs alone gives held-out margin inflation of 1.203 with an undefined lower limit and an upper limit of 1.828, and accuracy inflation of 1.299 with interval $[0.959,\ 1.584]$. The original ten benchmarks give 1.110 with interval $[0.686,\ 1.407]$ and 0.987 with interval $[0.310,\ 1.320]$ at that size.

Table 11. Median margin inflation over subsets of the original ten benchmarks, with the share of subsets whose nominal 95% interval excludes one, at 530M and 750M, at 530M, 750M, and 1B, at 530M alone, and across all five sizes. Benchmark counts with more than 60 subsets use 60 sampled without replacement, and subsets whose bootstrap degenerates are omitted from that share. The two-size rows were fixed before any 750M held-out score existed, the 530M rows came after the 530M interim result, and the three-size rows came after the full-scope result. These intervals use 1,999 wild draws, against 4,999 for the pre-specified test.

Benchmarks	2	3	4	5	6	7	8	9	10
530M and 750M median	1.010	1.143	1.213	1.288	1.142	1.206	1.210	1.259	1.316
530M and 750M share	0.37	0.53	0.55	0.53	0.37	0.33	0.20	0.10	0.00
530M, 750M, and 1B median	1.014	1.109	1.172	1.227	1.097	1.156	1.163	1.196	1.240
530M, 750M, and 1B share	0.36	0.48	0.55	0.50	0.42	0.45	0.58	0.80	1.00
530M median	1.019	1.133	1.244	1.317	1.296	1.368	1.377	1.439	1.514
530M share	0.33	0.58	0.67	0.80	0.78	0.87	0.91	0.90	1.00
All sizes median	1.012	1.116	1.176	1.228	1.100	1.160	1.160	1.200	1.244
All sizes share	0.38	0.53	0.57	0.62	0.73	0.82	0.98	1.00	1.00

C.23 Recipe decisions For each size below 1B, we formed every pair of the 25 recipes and averaged each recipe's aggregate score over its three runs and both halves. We called a pair when its gap exceeded two standard errors, $\sqrt{2/R}$ times the aggregate seed standard deviation, and we estimated that deviation either under independence or with the covariance, in both cases without the pair's two recipes. The same pair at 1B supplies the reference, and it counts as resolved when its gap exceeds two corrected standard errors, again without the pair. Table 12 reports wrong-sign calls among calls on resolved pairs.

A recipe bootstrap of 2,000 draws resamples the pair outcomes and doesn't re-estimate the standard deviations. Its intervals understate the uncertainty in those estimates. The share of resolved pairs that each rule calls correctly lies between 0.75 and 0.87 under both rules.

Table 12. Wrong-sign calls against the 1B ordering, as a share of calls on resolved pairs, under the independence and corrected rules. The call columns count calls on resolved pairs only, and the difference is the independence rate minus the corrected rate, with its recipe-bootstrap interval.

		Calls		Wrong-call rate		
Score	Size	Indep.	Corrected	Indep.	Corrected	Difference
Margin	150M	180	179	0.056	0.050	$[0.000, 0.022]$
Margin	300M	184	178	0.060	0.051	$[-0.002, 0.049]$
Margin	530M	181	169	0.061	0.059	$[-0.016, 0.022]$
Margin	750M	173	169	0.029	0.030	$[-0.004, 0.000]$
Accuracy	150M	217	215	0.069	0.070	$[-0.003, 0.000]$
Accuracy	300M	214	212	0.051	0.052	$[-0.003, 0.000]$
Accuracy	530M	223	218	0.067	0.064	$[-0.004, 0.024]$
Accuracy	750M	201	200	0.045	0.045	$[-0.002, 0.001]$

C.24 Null seed pairs in PolyPythias Since a wrong call in Table 12 can be a genuine rank change, the table gives no false-positive rate. PolyPythias gives one for the same threshold with deviations from one configuration's seeds, since its nine seeds at each size rerun one recipe and any two disjoint sets of them differ only by seed. On the 4,755-item transfer reductions of the 45 runs, we applied the pair rule above to each of the 36 pairs of single seeds at each size, with both standard deviations estimated from the other seven seeds, and to each of the 840 pairs of disjoint seed triples, with the deviations estimated from the remaining three. Two known standard errors give a nominal rate of 0.046. Pooled over the five sizes, both rules call 24 of 180 single-seed margin pairs, a rate of 0.133. On accuracy the independence rule calls 0.204 of 142 pairs with a defined deviation and the corrected rule 0.184 of 152.

With triples the margin rates are 0.193 under independence and 0.218 corrected. The accuracy rates are 0.296 of 2,900 defined pairs and 0.217 of 3,310. The largest single-seed margin rates are 0.333 under independence at 410M and 0.222 corrected at 70M. When all nine seeds estimate the deviations, pair included, the corrected rule calls 0.050 of single-seed margin pairs against 0.122 under independence, and 0.104 against 0.215 on accuracy. The held-out designs estimate each deviation on seven or three runs, and t references with six and two degrees of freedom put 0.092 and 0.184 of their mass beyond two standard errors. The check uses one configuration per size, the pairs at one size share seeds, and seed-jackknife standard errors of the per-size margin rates reach 0.222, so its rates are exploratory.

C.25 Split against replicate standard deviations The replicate standard deviation of the full-bank battery average needs no split. We asked which decisions on the released runs change when it replaces the split estimate. We computed both with the same estimator, with the full-bank scores as both halves for the replicate value, and applied three rules with the same two-standard-error logic to the full battery and to each single-benchmark removal on both scores. The pair rule above ran at all five sizes. A size-step rule calls a recipe's change between adjacent sizes when it exceeds two standard errors of the difference, and a seed-count rule asks for $\lceil 2(2\sigma/\delta)^2 \rceil$ runs with δ at the 25th and 50th percentiles of absolute pair gaps.

The replicate estimate never made a call that the split withheld. Of 16,500 pair decisions per score the split makes 60 extra margin calls, 11,705 against 11,645, and 382 extra accuracy calls, 12,861 against 12,479. On the full battery the extra calls number 4 on margins and 25 on accuracy. The battery without BoolQ has the most on accuracy, 61.

Below 1B, 198 of the 318 extra calls have a 1B ordering that clears two standard errors under both estimates, and 171 of those 198 agree with it. The size-step rule differs in 17 of 2,200 decisions, each an extra split call. The seed-count rule asks for more runs under the replicate estimate in 34 of 220 cases, 26 of them on accuracy. On the full population the replicate deviation exceeds the split

one in every battery, by about one percent on margins and 6 to 24 percent on accuracy. It's the more cautious of the two, and the decisions that change with the estimate are mostly on accuracy.

C.26 Runs per configuration The seed-count simulation draws populations from the default item-level simulator, with 125 configurations, the released item counts, its default item noise, and an equicorrelated seed covariance at the observed effective coefficient $\bar{r}_E$, 0.061 on margins and 0.018 on accuracy. It sweeps two, three, four, five, six, and eight runs per configuration with 300 replicates at each count and applies the primary wild interval with 999 draws. On margins the interval excludes one in 0.86 of replicates at two runs and in 0.997 at three, with median widths of 0.295 and 0.208. On accuracy the share rises from 0.147 at two runs through 0.267 at three and 0.527 at five to 0.803 at eight.

Equicorrelation spreads the excess evenly across benchmarks, whereas the observed covariance concentrates in BoolQ. These shares describe a battery with the same $\bar{r}_E$ and not this battery's structure. At the accuracy battery's own noise the same three-run design detects 1.078 in 0.111 of replicates and in 0.103 against the measured matrix, so the accuracy shares here overstate its power.

D Implementation details

D.1 Data streaming

The implementation streams 122.9 GB of gzipped JSONL tarballs from the Hugging Face release. It retains the item identifier, correct-answer margin, and accuracy bit for the ten traits at the common checkpoint. Each run contains 37,682 items and approximately 30 MB of compressed output. Reduction of 375 runs produces approximately 48 MB of NumPy archives, with one array per run.

A frozen master seed generates a balanced permutation of item order within each trait. The split uses that permutation instead of a hash of item identifiers, and all runs within an arm reuse the same trait-specific assignment. The PolyPythias subsample therefore has its own split and doesn't share an item-hash split across arms. A regeneration of the split requires the recorded item ordering and randomisation specification, and it doesn't need a download of the full tarballs.

D.2 Interval calculation

Derived quantities throughout the paper are computed from unrounded estimates. Recomputation from the printed three-decimal values can differ in the final digit.

We state the source calculation in terms of the squared ratio,

$$\widehat{\theta} = \frac{\sum_c T_c}{\sum_c U_c}, \qquad \widehat{\Lambda} = \sqrt{\widehat{\theta}}.$$

For each recipe g among $G = 25$, the linearised contribution and standard error in Equation 18 are

$$\psi_g = \frac{\sum_{c \in g}(T_c - \widehat{\theta} U_c)}{\sum_c U_c}, \qquad \widehat{\mathrm{SE}} = \left(\sum_{g=1}^{G} \psi_g^2 \right)^{1/2}. \tag{18}$$

This standard error belongs to the squared-ratio scale used for studentisation. A reported standard error on the Λ scale instead requires the corresponding delta-method conversion.

For draw b, independent recipe signs satisfy $v_{bg} \in \{-1, +1\}$ with equal probabilities, every configuration takes the sign assigned to its recipe. The pseudo-numerator becomes

$$T_c^* = \widehat{\theta} U_c + v_{bg(c)}(T_c - \widehat{\theta} U_c).$$

The denominator remains fixed at the observed $\sum_c U_c$. We recompute $\widehat{\theta}_b^* = \sum_c T_c^* / \sum_c U_c$ and the bootstrap standard error from the residuals $T_c^* - \widehat{\theta}_b^* U_c$. The bootstrap statistic is $t_b^* = (\widehat{\theta}_b^* - \widehat{\theta})/\widehat{\mathrm{SE}}_b^*$, and the source calculation uses 4,999 draws with bootstrap seed zero. The CPU kernels that carry the estimation, bootstrap, and permutation work ran on Python 3.12.13 with NumPy 2.0.2 under Linux. The scoring and calibration kernels ran on GPU cards. The draws come from NumPy's default generator, whose stream is tied to that version.

We compared the split with an analytic correction that keeps the full bank on both sides and subtracts $\sum_j v_{cj}/K^2$ from T_c and U_c, where v_{cj} comes from the item-by-run interaction mean square inside each group of benchmark j. That correction assumes item-by-run noise independent across the items of a group, which the split doesn't need. On margins it gives 1.246 $[1.144,\ 1.340]$ against the split's 1.244 $[1.143,\ 1.338]$, and on accuracy 1.082 $[0.994,\ 1.162]$ against 1.078 $[0.993,\ 1.157]$. Without BoolQ it gives 1.792 $[1.699,\ 1.878]$ against 1.786 on margins and 1.595 $[1.450,\ 1.721]$ against 1.558 on accuracy.

In the same order the analytic wild interval is 1.005, 1.025, 0.954 and 0.986 times as wide as the split's, since recipe-level seed variation sets the width. Item noise makes up 0.037, 0.186, 0.092 and 0.578 of the full-bank diagonal on these four batteries, where the uncorrected full-bank values are 1.237, 1.067, 1.734 and 1.285. The fixed-bank and split targets nearly coincide on full-battery margins and separate most on accuracy without BoolQ.

With q_α the empirical quantile of these statistics, the interval on θ is

$$\left[\widehat{\theta} - q_{0.975}\widehat{\mathrm{SE}}, \widehat{\theta} - q_{0.025}\widehat{\mathrm{SE}}\right].$$

Square-root transformation gives endpoints on the inflation scale when both endpoints are nonnegative. Monotonicity preserves the ordering of available endpoints, but it doesn't establish the original interval's coverage.

A draw with zero bootstrap standard error has no finite studentised statistic and is discarded. The routine requires at least $\max(100, 0.9B)$ valid draws before it reports an interval. A negative lower squared-ratio endpoint is recorded as missing and isn't truncated, which explains the absent 1B margin lower endpoint in Table 6. The configuration percentile bootstrap instead resamples all 125 configurations with replacement and recomputes both sums and ignores recipe membership.

The planning document specified wild cluster bootstrap-t inference with $t(16)$ critical values for seventeen estimation recipes. The realised analysis uses empirical bootstrap quantiles over all 25 recipes, following the failure to implement the planned split. The analysis record states that a $t(G-1)$ variant accompanies the bootstrap implementation. These alternatives differ in more than a change of cluster count.

D.3 Practical considerations

Within-configuration centring removes the shared performance mean before we form cross-half products. With only three runs, raw products would include a large contribution from mean competence and wouldn't measure covariance of deviations. The contrast representation in Appendix D.7 enforces that centring at the estimator input.

The permutation procedure reassigns run labels independently within each trait. One joint permutation of all traits preserves the cross-trait pairs and therefore doesn't generate this independence null. Identity permutations remain valid members of the randomisation distribution and shouldn't be excluded. For three runs, independent identities across all K traits have probability 6^{-K} within one configuration.

Gzipped tarball members require sequential reading from their starts. The reduction therefore extracts the required fields on one pass, since an omitted field requires another complete stream. The default-versus-auxiliary step mismatch at 530M and 750M motivates the auxiliary-two-versus-auxiliary-three contrast, with checkpoint agreement checked separately for both auxiliary runs.

The pipeline lists the 375 checkpoint directories once, applies exponential backoff to rate limits, and caches the listing. Before estimation, it regenerates the common-step table from that cache and compares it with the initial version. This check can detect changes in the recorded step selection, but a cached listing alone can't detect new remote changes without a refreshed comparison. Table 13 carries the planned compute budget for both arms.

D.4 Compute

The primary run used one Kaggle CPU core for Arm 1 and reported 6.0 hours for estimation and null checks. Four sensitivity tasks took another 1.6 hours, with gain calibration and fifty re-splits instead

Table 13. Compute budget reported in the planning document for both analysis arms. These figures describe the plan and aren't measured costs for every listed operation.

Operation	Budget	Scope
Arm 2 held-out bits per byte	6.0 GPU hours	27 runs, 4,755 items
Calibration simulations N1, N5, N6	1.0 GPU hours	2,000 replicates, six effects
160M checkpoint check	0.5 GPU hours	Adjacent checkpoints
Contingency	2.5 GPU hours	Reserved capacity
Total GPU	10.0 GPU hours	Planned total
Streaming and reduction	18 CPU hours	122.9 GB, one pass
Estimation, bootstrap, permutation	7 CPU hours	Includes 200 planned splits

of the 200 budgeted, and the exact recipe sweep took thirteen seconds. The later 160M checkpoint check took 423 seconds of scoring on one T4. The held-out test scored 205 of its 225 runs on Kaggle T4 cards and the last 20 at 1B on three RTX 3090 cards in 6,068 seconds, after the weekly quota ran out. Arm 1 performs no forward passes and reports no GPU use.

The transfer procedure streams 25 monolithic tarballs, 122.9 GB in total, and keeps peak disk use below 10 GB. Arm 2 performs teacher-forced fp16 scoring with zero-shot contexts and no exemplars. Most items use one context, but WinoGrande uses one per choice. Reported context lengths include 199 tokens for ARC-Challenge and 937 for BoolQ.

The budget assumed two 16 GB cards, but the reported scoring jobs instead received one 46 GB L40S each on a Slurm cluster. Each of 27 jobs finished in under two minutes, with aggregate reported use below one card-hour against the budgeted six. Gold-token scoring materialises a float32 log-softmax tensor over a 50,304-token vocabulary. A 30,000-token batch requires approximately 6 GB for that tensor alone.

A configuration rescore fit on a 23 GB card at the planned token budget, but the 16 GB T4 checkpoint check used 6,000 tokens. A change to the budget changes batch grouping and can affect floating-point results even when the mathematical scoring rule stays fixed. The transport rescore changed hardware and token budget together and shifted inflation by 0.6 percent. CPU handles the reduction and statistical analysis, with model scoring and calibration kernels as the reported exceptions.

D.5 Release provenance

The null checks examine mechanisms that can affect covariance estimates, but they don't independently establish the provenance of the three released runs. Differences in tokeniser builds, harness revisions, or other training conditions could contribute to the default-versus-auxiliary contrast if those conditions weren't held fixed. The auxiliary-run contrast also relies on documentation that the relevant settings match between its two runs.

The step table and disagreement checks expose some inconsistencies but don't replace a release-side audit. Attribution of the observed contrasts to the documented seed changes therefore depends on the release documentation. This limitation applies independently of the estimator's behaviour on simulated or permuted data.

D.6 Sensitivity to GPU architecture

The primary Arm 2 scores use one card type. A separate Turing-versus-Ada comparison on one configuration reports mostly identical item scores, a mean absolute margin difference of 1.0×10^{-3}, and a largest difference of 3.5×10^{-2}. Relative to a margin standard deviation of 0.391, the mean difference has ratio 2.6×10^{-3}.

Eighteen of 4,755 accuracy bits change, or 0.38 percent, with a median absolute margin of 1.3×10^{-3}, and battery accuracy changes from 0.430074 to 0.430494, a difference of 4.2×10^{-4}. That difference is approximately eight percent of the reported accuracy seed standard deviation. This single comparison shows sensitivity at that scale, but it doesn't establish a universal portability threshold for fp16 kernels.

Arm 1 uses released per-item outputs and doesn't recompute forward passes. This study doesn't add the same hardware variation to those scores. Shared hardware across Arm 2's 27 runs reduces one source of variation but doesn't independently validate accuracy across architectures.

D.7 CENTRING THROUGH RUN CONTRASTS

Raw cross-half products contain squared configuration means, and the analysis record reports that these can inflate the ratio by roughly three orders of magnitude. The described implementation projects scores onto an orthonormal basis of mean-zero run contrasts. Each basis row annihilates the all-ones vector, and raw and centred inputs are therefore equivalent under projection. Runtime checks test row sums and invariance of $\sum_c T_c$ under a shared configuration shift on both halves, and they catch centring errors but don't test measurement or exchangeability assumptions.