Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Mind the Gap: Explicit Cross-Benchmark Calibrations for Estimating Missing LLM Evaluation Scores

Patrik P. Süli

suli.patrik@uni-obuda.hu

Obuda University

György Eigner

Obuda University

Roland Hollós

HUN-REN Centre for Agricultural Research

Research Article

Keywords: Large language models, Benchmark evaluation, Missing-score estimation, Cross-benchmark calibration, Matrix completion, Evaluation harness

Posted Date: October 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11552635/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Mind the Gap: Explicit Cross-Benchmark Calibrations for Estimating Missing LLM Evaluation Scores

Patrik P. Süli[1,2*], György Eigner[2,3] and Roland Hollós[4]

[1*]Doctoral School of Applied Informatics and Applied Mathematics, Obuda University, Budapest, Hungary.
[2]Biomatics and Applied Artificial Intelligence Institute, John von Neumann Faculty of Informatics, Obuda University, Budapest, Hungary.
[3]Physiological Controls Research Center, Obuda University, Budapest, Hungary.
[4]HUN-REN Centre for Agricultural Research, Brunszvik u. 2., Martonvásár, 2462, Hungary.

*Corresponding author(s). E-mail(s): suli.patrik@uni-obuda.hu;

Abstract

Most language models are evaluated on only a few benchmarks: in our compilation of 401 models and 146 benchmark versions, the benchmark test coverage is only 8.3% on the most frequently used benchmarks. We present *benchgap*, a pipeline that estimates the missing scores without running any evaluations. It calibrates each benchmark against each same-capability benchmark on the models measured on both, fitting six monotone curve families and keeping the one with the lowest leave-one-out cross-validated error; multivariate mappings combine several sources. Calibrations never cross capability groups, must rise and never extrapolate beyond one fitted-range width; estimates never feed further estimates, and each carries its own cross-validated error and confidence level. In an end-to-end holdout of 15% of the measured scores, the pipeline fills 71% of the masked cells at 4.6 percentage points (pp) mean absolute error (MAE), 23% lower than a rank-2 matrix-completion baseline on the same cells; on the cells it leaves empty, the baseline's MAE is 11.8 pp. Realized MAE rises with the confidence level (3.1, 5.1 and 7.5 pp from high to low), but the realized root-mean-square error exceeds the claimed error labels, which are selected minima, by a factor of 1.4–1.7. We also quantify the *harness tax*, the disagreement between evaluation setups: on verified same-item-set pairs it is 8.7 pp on agentic benchmarks against

2.3 pp on tool-free ones, and enabling tools on Humanity's Last Exam raises the scores of all 11 models measured both ways, by 10.0 pp on average. Estimates are updated daily at https://benchgap.net.

Keywords: Large language models, Benchmark evaluation, Missing-score estimation, Cross-benchmark calibration, Matrix completion, Evaluation harness

1 Introduction

New language models are released faster than they are evaluated on the benchmarks in common use. Each provider reports scores on benchmarks of its own choosing, each evaluation service runs its own subset of models, and the union of all forms a sparse model × benchmark matrix whose cells are scores. For the 401 models and 146 benchmark versions we track, only 8.3% of the cells are measured, and the median benchmark version has measured scores for just 16.5 models (Figure 1). Models compared across leaderboards are therefore compared on tables in which more than 90% of the cells are empty, and every analysis that needs a complete matrix (capability profiling, model selection, meta-evaluation) starts with missing data.

A growing research line predicts a model's benchmark scores without running every evaluation: matrix completion over the score matrix (Zeng and Papailiopoulos 2026), scaling laws over latent skill factors (Polo et al. 2024; Ruan et al. 2024), psychometric factorization from a few tasks (Maimon et al. 2025), item-level coresets (Bowyer et al. 2026; Kipnis et al. 2024), and prediction from a text description of the benchmark before any evaluation exists (Park et al. 2025). These methods show that benchmark scores are largely redundant, in that a few latent factors per model account for most of the matrix. They share a design: one global method, fitted in a latent space, evaluated by one aggregate error over held-out cells.

benchgap takes the complementary position. It estimates the same missing cells with explicit calibrations: every stored calibration names an observed source benchmark and an observed target benchmark, is fitted by least squares with two to four parameters, and carries its own leave-one-out cross-validated error. Each estimate can be traced to its source score, curve and fitted parameters through the application programming interface (API). The cost of this explicitness is that no information is shared across the whole matrix, and this paper measures that cost: on a masked hold-out of measured cells, the explicit pipeline has 23% lower mean absolute error (MAE) than a rank-2 matrix-completion baseline on the cells it fills (16% lower than the best rank of the sweep), and on the cells the pipeline leaves empty the baseline's error is double its error on the filled ones (Section 5.3).

Concretely, this paper makes the following contributions:

1. **A transparent gapfilling pipeline** (Section 4): per-pair rising calibrations with leave-one-out model selection, a quality gate that leaves a cell empty when no calibration reaches $R^2 \geq 0.3$ and a leave-one-out (LOO) error of at most 15 percentage points (pp), and the five constraints of Section 4.3.

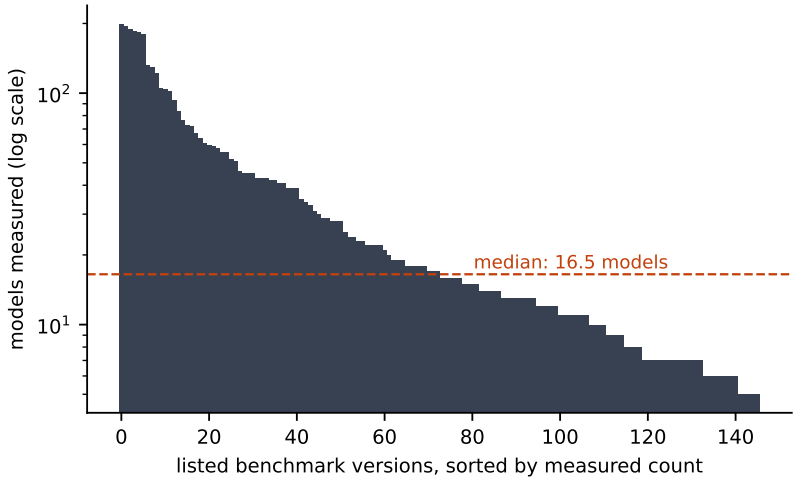


Fig. 1 The gapfilling problem. Of the 401 × 146 cells in the listed model × benchmark matrix, only 8.3% are measured; the median listed benchmark version has scores for 16.5 models

2. **Multivariate mappings** (Section 4.2): each benchmark predicted from several others together, with feature selection done by an elastic net whose lasso part zeroes uninformative sources, or by a greedy forward search.
3. **An end-to-end evaluation** (Section 5.3): the whole pipeline refitted with 15% of measured cells masked, testing whether the pipeline's published errors and confidence levels match its realized errors, against a rank-2 matrix-completion baseline.
4. **The harness tax** (Sections 4.5 and 5.4): a measured-scores-only quantification of how much evaluation setups disagree about the same models on the same benchmark. To our knowledge, no published study measures this disagreement per model across benchmark families; the daily-refreshing data makes such a measurement possible.
5. **A running system**: the pipeline is MIT-licensed, refreshes its data daily, and publishes every score, calibration and provenance record through a free JSON/CSV API with an OpenAPI description (benchgap contributors 2026).

The paper is a snapshot of a live system: all numbers refer to the data retrieved on 8 October 2026. The pipeline is public; the live API refreshes daily, so it describes the present snapshot rather than this paper's (Section 8).

2 Related work

Closest to our problem is the matrix-completion line. Zeng and Papailiopoulos (2026) show that the model × benchmark score matrix is nearly rank-2 and complete it in logit space (BenchPress). Maimon et al. (2025) factor scores into low-rank psychometric factors and profile a model from a subset of its tasks. benchgap keeps the calibration local and explicit: one curve per ordered benchmark pair, with its own error, and

no transfer across capability groups. The global methods impute between any two benchmarks; we leave cross-capability cells empty.

In the scaling-law line, Ruan et al. (2024) regress downstream benchmark performance on a latent capability variable anchored to compute, and Polo et al. (2024) fit scaling laws over latent skills within and across model families. These predict models that have not been trained yet and need training metadata. benchgap maps a model that exists from its measured scores alone (which is all that closed API models publish) and needs no compute information. Owen (2024) studies benchmark predictability against five orders of magnitude of training compute; our inputs are scores only.

At the item level, Kipnis et al. (2024) distill benchmarks to a sparse ($\sim$3%) subset of items, and Bowyer et al. (2026) recast efficient benchmarking as feature selection plus regression over items. Our multivariate stage is the benchmark-level analogue: the lasso selects benchmarks the way they select items. benchgap works from published aggregate scores, so it needs no access to benchmark items at all.

Park et al. (2025) predict a model's score from a redacted description of the task, before any evaluation exists. benchgap predicts after a model has some measured scores; the two ends of the pipeline are complementary.

Score prediction has itself come under scrutiny: Pacchiardi et al. (2025) benchmark the predictors themselves, and Zhang et al. (2025) show systematically that score-prediction methods fail on models unlike their training set. The constraints of Section 4.3 and the confidence demotions of Section 4.4 are built around that failure mode, our holdout evaluation (Section 5.3) tests whether they have the intended effect, and Section 6 returns to the failure mode as the shared limit of the whole research line.

3 Data

Measured scores come from the public compilation of BenchLM.ai (BenchLM.ai 2026), which aggregates evaluation results from three sources: the model providers' own published numbers (*published*), Artificial Analysis (*AA*) and VALS AI (*VALS*). Each source runs its own evaluation *harness*, the software and protocol that execute a benchmark. A *benchmark version* is one benchmark release as reported by one source (e.g. Terminal-Bench 2.1 by VALS); successive releases of a benchmark (Terminal-Bench 2.1 and 4.0) are distinct versions too. Every score is stored as a fraction of the maximum, with the retrieval date and the source as provenance; measured scores are never altered.

Each benchmark belongs to one of ten capability groups (agentic terminal, agentic tools, coding, math, knowledge and reasoning, instruction following, multilingual, vision and documents, long context, general), assigned from BenchLM's category labels, with name rules for terminal and long-context benchmarks; a benchmark with no matching category falls into general. Benchmarks are calibrated only against benchmarks of the same capability (Section 4.3).

Table 1 summarizes the snapshot analysed in this paper. The site *lists* the benchmark versions that have at least one estimate and at least ten scores of any kind; everything the pipeline stores but the site does not list is *tracked*.

Table 1 The data snapshot (retrieved 8 October 2026). Of the 14,314 estimates, 14,311 fall inside the listed matrix; the tracked set additionally holds 1,293 measured scores on versions or models the site does not list.

	listed	tracked
models	401	520
benchmark versions	146	359
measured scores	4,887	6,180
estimated scores	14,311	14,314
stored fits, over all tracked versions		
univariate calibrations		1,147
multivariate mappings		21
cross-capability, univariate (analysis only)		6,784
cross-capability, multivariate (analysis only)		117

Table 2 The six candidate curve families and how often each is selected (lowest leave-one-out error) over the 1,147 stored calibrations.

family	equation	parameters	selected
Hill	$y = y_0 + (a - y_0)\, x^n/(k^n + x^n)$	4	273
linear	$y = ax + b$	2	248
Michaelis–Menten (MM)	$y = V_{\max}\, x/(K + x)$	2	221
offset logistic	$y = y_0 + (a - y_0)/(1 + e^{-k(x - x_{\mathrm{mid}})})$	4	218
inverse MM + offset	$y = K(x - y_0)/(y_0 + V_{\max} - x)$	3	94
MM + offset	$y = y_0 + V_{\max}\, x/(K + x)$	3	93

4 Method

4.1 Univariate calibrations

For every ordered pair of same-capability benchmark versions with at least five models measured on both, six monotone curve families (Table 2) are fitted by least squares on the models measured on both, with SciPy's `curve_fit` (Levenberg–Marquardt for the unbounded linear form, trust-region reflective with parameter bounds for the others; at most 20,000 function evaluations, 40,000 for the inverse form). Each family starts from one data-driven guess (e.g. the observed minimum target score for the lower asymptote and slightly above the observed maximum for the upper one); the bounds keep lower asymptotes in $[0, 1]$, upper asymptotes in $[0, 1.2]$ and $V_{\max}$ in $[0, 2]$, and predictions are clipped to $[0, 1]$. A family whose fit fails to converge is dropped for that pair. The two versions may come from different sources; we quantify the consequences in Sections 5.3 and 5.4. The saturating families exist because the typical relationship between two benchmarks is concave in the source: gains on an easier benchmark flatten out while a harder one keeps discriminating.

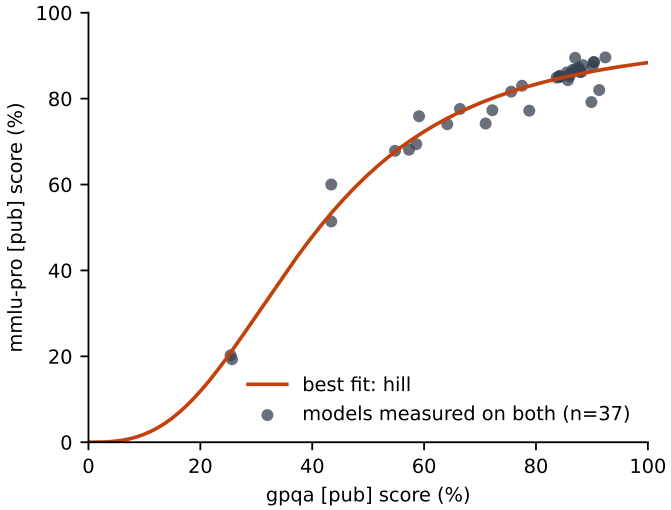


Fig. 2 A stored univariate calibration: GPQA (published) → MMLU-Pro (published), a Hill curve fitted on the 37 models measured on both ($R^2 = 0.97$, leave-one-out error 3.0 pp)

Only rising fits are kept. Every family is monotone, but the linear, Hill and logistic forms can also fit a falling curve (a negative slope, an asymptote below the baseline). Such a fit claims that a higher score on one benchmark predicts a lower score on the other, which in practice means that too few models cover part of the range, not that the two benchmarks are anti-correlated. For instance, a falling Hill curve from Terminal-Bench 2.1 to 4.0, fitted on models scoring 81–93% only, would have estimated a small model scoring 8.6% on 2.1 at about 100% on 4.0. Falling fits are therefore discarded before selection: a linear fit needs a positive slope, a Hill or logistic fit an asymptote above its baseline, a Michaelis–Menten fit a positive $V_{\max}$. A pair is calibrated by its best rising curve or not at all.

For each pair, the curve with the lowest *leave-one-out cross-validated* (LOO) error is selected: each model is held out in turn, the curve is refitted without it, and the held-out score is predicted (above 40 models, 40 evenly spaced models are held out). The resulting root-mean-square error (RMSE) in percentage points is the error attached to the calibration and to every estimate made through it. If a held-out refit fails to converge, the family has no LOO error and is ranked after the families that have one; no stored calibration in the snapshot lacks an LOO error. Because the same LOO errors also select the family per pair and the mapping per cell, they are a selected minimum and therefore optimistic; Section 5.3 measures by how much, end to end. A pair keeps no calibration unless its best curve reaches $R^2 \geq 0.3$ and an LOO error of at most 15 pp; such cells stay empty. Figure 2 shows a typical surviving fit.

4.2 Multivariate mappings

A single source benchmark often predicts the target with large error, so for each target benchmark a multivariate mapping predicts it from several same-capability

benchmarks at once. Feature selection and the final functional form are chosen in two separate steps, both by the same LOO error.

For feature selection, the candidate sources are the same-capability versions that share at least five models with the target, ranked by their best single-source LOO error for that target and then by overlap. Two searches run and the one with the lower LOO error is kept: (i) an elastic net fitted over a pool of up to the eight best-ranked candidates, whose lasso part sets the coefficients of uninformative sources to zero; the penalty is chosen by LOO error on the grid $\lambda/\lambda_{\max} \in \{0.5, 0.25, 0.1, 0.05, 0.02, 0.01, 0.005\}$ and L1 ratio $\in \{1.0, 0.9, 0.5\}$, and at least one source must survive; and (ii) a greedy forward search that, at each step, adds the remaining candidate that most reduces LOO error, and stops at three sources or when no addition reduces it by more than 0.01 pp. A pool or mapping on k sources requires at least $5 + 2k$ models measured on the target and every source; a candidate that would break this is skipped. Feature selection and the penalty are chosen once, on all models; the LOO error then refits the coefficients (or, for the Michaelis–Menten form, the whole curve) with the selected sources fixed, so it is optimistic in the same way as the univariate selection.

The functional form is chosen next: on the features each search selects, two families are compared by LOO error: the linear elastic net with non-negative coefficients (a positive lasso, solved by coordinate descent), so that a higher source score never lowers the estimate,

$$\hat{y} = \beta_0 + \sum_{j \in S} \beta_j x_j, \qquad \beta_j \geq 0, \tag{1}$$

and a multivariate Michaelis–Menten curve that maps the sources, combined into a non-negative weighted index, through the same saturating shape as the univariate curves:

$$s = \sum_{j \in S} w_j x_j, \qquad \hat{y} = y_0 + V_{\max} \frac{s}{K + s}, \qquad w_j \geq 0, \tag{2}$$

fitted with `curve_fit` under the same bounds as the univariate curves and $w_j \in [0, 10]$, with the weights then normalized to sum to one; it is tried only when at least two sources are selected.

A multivariate mapping is stored only if it passes the same quality gate as a univariate one and has a lower LOO error than the target's best univariate calibration. A lasso-selected single source may be stored (on little overlap its shrinkage can give a lower LOO error than every univariate curve), but a one-feature nonlinear fit is not, since the univariate calibrations already cover that case. Figure 3 shows the largest stored mapping.

4.3 Filling a gap

From here on, *mapping* covers both univariate calibrations and multivariate mappings. For a model missing a score on some benchmark, every stored mapping into that benchmark from a benchmark the model was measured on is eligible; the eligible mapping with the lowest LOO error is used. A multivariate mapping is preferred when the model is measured on all of its source benchmarks and its error is lower; a

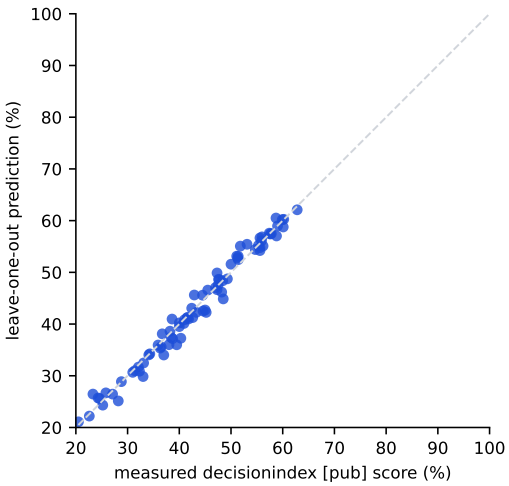


Fig. 3 The largest stored multivariate mapping: Decision Index (published) predicted from its five sub-index scores (Arts, Knowledge, Language, Retrieval, Tools) by an elastic net, 112 models measured on all six. Each point is the prediction made for that model by a fit that did not see it

model missing one source falls back to the univariate path. The pipeline enforces five constraints:

(i) **No estimate recursion.** Every input to an estimate is a measured score; an estimate never feeds another estimate.

(ii) **Capability gating.** Every calibration is within-capability, so no estimate crosses a capability boundary: a model never evaluated on vision keeps that cell empty instead of receiving an estimate from text benchmarks.

(iii) **Rising calibrations only** (Section 4.1).

(iv) **Bounded extrapolation.** A calibration fitted on source scores in $[x_{\min}, x_{\max}]$ is used only on source scores within one range width $w = x_{\max} - x_{\min}$ of it, in $[x_{\min} - w, x_{\max} + w]$. Beyond that interval the curve is not constrained by data, and the cell stays empty. Inside that reach but outside the fitted range the estimate is made and flagged as an extrapolation (Section 4.4). A multivariate mapping applies the rule to each of its sources.

(v) **Per-cell error and confidence level.** Every estimate carries the LOO error of the mapping that made it and a confidence level (Section 4.4).

Each stored estimate records its full provenance: source benchmark, source score, curve family, parameters, and whether the source score lay outside the mapping's fitted range. An estimate is written into a specific target version, so it refers to that version's measurement setup: it is a prediction of what that source's harness would have measured. When source and target versions come from different sources, the curve, fitted on the models measured under both, absorbs the pair's average harness difference into its coefficients; the model-level part of that difference remains in the calibration's error (Section 5.3 measures its effect, Section 5.4 its size).

Table 3 Confidence levels for the 14,311 estimates in the snapshot.

level	rule	count	share
high	LOO error ≤ 5 pp and no warning	2,457	17%
medium	LOO error ≤ 10 pp, or high with one warning	4,929	34%
low	above 10 pp, or demoted to low by warnings	6,925	48%

Two further fits are computed for analysis only and never produce an estimate. Cross-domain predictability fits every cross-capability pair under the same gate, to show which capabilities predict which (for instance, how well coding scores predict agentic terminal scores). The multivariate view fits each benchmark from several benchmarks of any capability and plots every model's cross-validated prediction. Published estimates come only from the within-capability calibrations of Sections 4.1 and 4.2.

4.4 Confidence levels

Every estimate carries a confidence level so that low-confidence fills are visibly different everywhere they appear. The level starts from the LOO error (high for ≤ 5 pp, medium for ≤ 10 pp, low above) and each of three warnings then demotes it by one level, with low as the floor: *extrapolation* (the source score lies outside the mapping's fitted range, within its reach), *small sample* (the mapping was fitted on fewer than 8 models, so its own error estimate is noisy) and *uninformative fit* ($R^2 < 0.5$). An estimate with two or three warnings is therefore low regardless of how small its LOO error is. Table 3 shows the distribution over the snapshot; Section 5.3 tests whether these levels match realized error.

4.5 Measuring the harness tax

Every benchmark version in the database records its source: the provider's own published numbers, AA or VALS. The same benchmark measured by two sources disagrees about the same models, sometimes by tens of points. We call this disagreement the *harness tax* and measure it from measured scores only.

Benchmark versions are grouped into *families* of versions of the same benchmark. A hand-written registry lists the known families, each with a tier and an item-set alignment status set by hand. The tier is *agentic* (the harnesses' agent loop, scaffolding and tool use differ), *tool-free* (no agent loop; harness choice should matter least) or *protocol layer* (the versions differ only in evaluation protocol, e.g. tools on or off); the alignment status is *verified*, *needs audit* or *weak alignment*. In addition, versions whose names coincide after normalization (lower-casing, removing source prefixes such as `aa-` and `vals-`, and dropping variant tokens such as `notools`, `withtools`, `pro` and `v6`, also when attached to the name, as in `charxivnotools`) form auto-detected families; these have no tier (shown as *candidate* in Section 5.4) and weak alignment until audited. For every pair of versions sharing at least one model measured on both, we compute per-model deltas $\Delta = y_a - y_b$ with full provenance, and summarize each

pair by mean and median $|\Delta|$, the largest $|\Delta|$, the share of models differing by more than 10 pp, Kendall's τ-b between the two versions' rankings, a two-sided sign test on sign(Δ), which tests whether the differences have a consistent direction, and the directionality (the share of models agreeing with the majority sign). Individual deltas above 20 pp are flagged into an audit queue. The harness tax never feeds the estimates; it measures how far scores from one evaluation setup can differ from another's.

5 Results

5.1 Calibrations

The snapshot stores 1,147 univariate calibrations. The rising-only rule removed 32 pairs whose best fit fell; none of them had a rising curve that passed the quality gate. Four families are each selected for 218–273 pairs and the two offset Michaelis–Menten forms for 93–94 (Table 2). Saturating forms are selected where target scores cluster near the maximum while source scores remain spread out, the linear form on near-identical benchmarks, and the inverse Michaelis–Menten on pairs whose relationship is convex in the source, typically a hard target whose scores accelerate as an easier source improves. The median stored calibration has $R^2 = 0.71$ and a median LOO error of 5.1 pp; every estimate is published with its error. Calibration overlap is often small: 476 of the 1,147 calibrations (41%) were fitted on only 5–7 models (the gate requires at least 5; below 8 the small-sample warning applies), and 5,409 of the 14,314 estimates (38%) are made through such small calibrations, all of them flagged with the small-sample warning. A calibration may also pair versions from different sources, as 502 of 1,147 (44%) do; the harness tax of that pair then adds to the calibration's error.

The 21 stored multivariate mappings (17 elastic net, 4 multivariate Michaelis–Menten) have lower LOO error than the univariate calibrations: median 1.9 pp against 5.1 pp. Most are small (median 10 models measured on all sources), because the number of models measured on every source falls quickly as sources are added; the largest (Figure 3) uses 112 models and predicts a composite index from its five sub-indices, which is an easy, nearly definitional case. In the current snapshot 203 estimates (1.4%) are made through a multivariate mapping; the rest use the univariate path, either because no multivariate fit passed the gate or because the model misses one of the chosen sources.

5.2 Estimates

The pipeline produces 14,314 estimates from 6,180 measured scores, raising the listed matrix from 8.3% measured to 32.8% populated. Their confidence split (Table 3) is skewed toward low by design (48% of listed estimates): any warning demotes an estimate, and a demoted estimate is still published, with its flag. Much of this is structural: 6,238 estimates (44%) are flagged as extrapolations, mostly because saturating curves have a ceiling: several models above a mapping's fitted range receive the same estimate. The largest such saturation groups assign one score to 94 models on one benchmark and 66 on another; the ties are visible in the published matrix and API as identical, extrapolation-flagged values, ranked adjacently on the leaderboard.

The two rules of Section 4 that restrict calibrations at the edges of their data (rising fits only, and no estimate beyond one range width of the fitted data) remove estimates; they do not adjust them. Without them the same data yields 16,443 estimates, of which 373 are made through a falling calibration and 2,684 lie beyond the reach of their calibration; some of those cells fall back to the next-best calibration, and the net cost is 2,129 estimates (13%). The confidence levels had already marked 83% of the lost estimates as low and none as high.

5.3 End-to-end holdout evaluation

To measure the realized accuracy of the published estimates, we mask a uniformly random 15% of all 6,180 measured cells (927 cells), delete them from the database, empty every fitted table, rerun the full pipeline (`fit`, `multifit`, `gapfill`) on the reduced data, and score the masked cells against their held-out truth. Two baselines are evaluated on the same masked cells for reference. The first is a rank-2 matrix completion in logit space, following the near-rank-2 finding of the BenchPress family (Zeng and Papailiopoulos 2026): on the full 520×359 tracked matrix, scores are clipped to $[0.005, 0.995]$ and logit-transformed, missing cells are initialized to their version's mean, and 50 iterations of hard-impute singular value decomposition (rank-r reconstruction, observed cells reset) produce the prediction. A rank 1–5 sweep is also reported; its best rank is chosen on the masked cells, which favours the baseline. The second is the mean of the version's remaining measured scores. The experiment is repeated over three mask seeds (20261008, 20261009, 20261010); the seed-20261008 numbers are shown, and the conclusions are stable across seeds: overall MAE 4.7 ± 0.2 pp (standard deviations across the three seeds), coverage 70.3–71.3%, the confidence levels correctly ordered in every seed, and the pipeline's mean per-cell absolute error lower than the completion's in every seed (mean paired difference, pipeline minus completion, -1.1 ± 0.5 pp).

Table 4 and Figure 4 summarize the results. Realized error rises with the confidence level: 3.1 pp MAE for high, 5.1 for medium, 7.5 for low, and the same holds in RMSE (5.1 / 7.0 / 11.3). The realized RMSE, however, exceeds the claimed LOO RMSE by a factor of 1.69 (high), 1.42 (medium) and 1.60 (low), computed from unrounded values; the high level's realized RMSE (5.1 pp) exceeds the 5 pp threshold that defines it. The ordering of estimates by claimed error thus matches the ordering of realized error, while the absolute labels are too small. The tails are heavy: RMSE/MAE ratios of 1.4–1.6 against a Gaussian's $\approx$1.25, with 90th-percentile absolute errors roughly double the MAE (6.9 / 12.3 / 15.3 pp), and the largest errors lie beyond the 90th percentile: 99th-percentile errors of 21 / 20 / 31 pp and worst-case misses of 34, 22 and 70 pp. A high-confidence estimate can also be far off: the largest miss in the holdout's high level is +33.6 pp (a model predicted at 72.6% on a benchmark where it measured 39.0%), and the largest anywhere is +69.8 pp on a low-confidence, extrapolated fill (predicted at 100%, measured at 30.2%); that extrapolation lies inside the one-width reach, which the reach rule therefore does not stop. Each demotion warning is associated with higher realized error: extrapolated fills erred at 6.6 pp MAE against 4.3 pp for non-extrapolated ones ($n = 68$ vs. 593), small-sample mappings at 6.0 against 4.1 ($n = 151$ vs. 510), and weak-fit mappings at 5.9 against 4.5 ($n = 32$ vs. 629). (The warnings

Table 4 The masked holdout: 927 measured cells hidden, the whole pipeline refit on the rest. "Claimed" is the mean LOO RMSE of the mappings used; realized error is measured against the held-out truth, in the same metric on both sides; p90: 90th-percentile absolute error; —: no claimed error. The completion cannot predict the 6 masked cells of 5 models whose every measured score was masked; the pipeline leaves those empty too.

		error (pp)			
	n	claimed	MAE	RMSE	p90
pipeline, by confidence level					
high	310	3.0	3.1	5.1	6.9
medium	242	4.9	5.1	7.0	12.3
low	109	7.1	7.5	11.3	15.3
completion (rank 2), on the same cells					
high	310	—	4.3	6.0	10.1
medium	242	—	6.8	10.4	15.5
low	109	—	8.9	12.4	20.3
all 661 filled cells					
pipeline	661	4.4	4.6	7.1	10.8
completion, rank 2	661	—	5.9	9.0	14.0
completion, rank 3 (best)	661	—	5.4	8.3	13.1
per-benchmark mean	661	—	12.2	15.9	25.2
the 266 cells the pipeline leaves empty					
completion, rank 2	260	—	11.8	16.9	30.3

are also rare in the holdout relative to production: only 10% of the holdout fills were extrapolated, against 44% of the published estimates.)

The holdout is also easier than the published matrix. Random masking removes cells mostly from well-measured models that keep many source scores, so the holdout fills skew easy: 47% high confidence against 17% in the published matrix, and only 31% carried a warning against at least 44% (extrapolation alone) in production. Reweighting the per-level realized errors to the published confidence mix gives the error a published estimate should be expected to carry: about 5.9 pp MAE and 9.1 pp RMSE (RMSE pooled in mean square), against 4.6 / 7.1 on the unweighted holdout.

Half of the empty cells of the published matrix (51%), however, belong to models with at most two listed measured scores. To test such sparsely measured models directly, we draw 40 models at random (seed 20261008) from those with at least 6 measured scores and keep only k of their scores, chosen at random ($k = 1$ and $k = 2$), masking the rest; all other models keep everything, so the calibration base stays realistic. Most cells of sparse models stay empty: with a single measured score, the pipeline estimates only 130 of 701 remaining cells (19%); with two, 173 of 661 (26%), against 71% on the random mask. On the cells it fills, MAE is 5.3 pp at $k = 1$ (130 cells) and 4.9 pp at $k = 2$ (173 cells); the random-mask result (4.6 pp) is on a different set of cells. The confidence levels stay correctly ordered (3.7 / 5.0 / 9.2 at $k = 1$; 3.0 / 4.8 / 7.1 at $k = 2$), and on the same sparse cells the completion baseline erred at 8.2 and

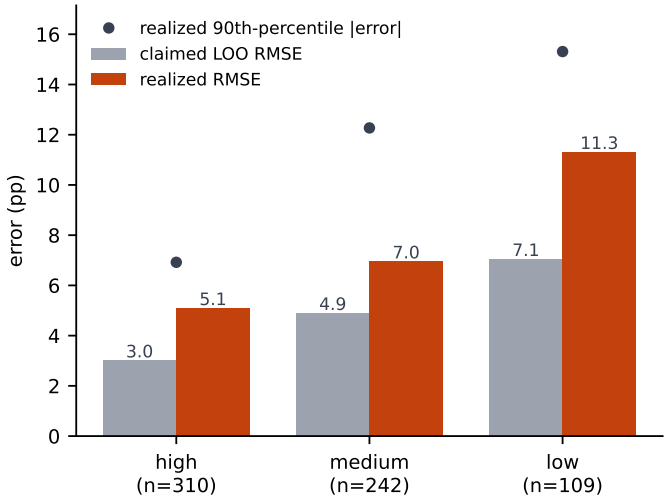


Fig. 4 Claimed versus realized error by confidence level on the masked holdout, both sides in RMSE; dots: 90th-percentile absolute error

7.9 pp (12.0 and 11.2 on the cells the pipeline leaves empty). Extrapolation is rare in these sparse holdouts (2 of 130 fills at $k = 1$, 13 of 173 at $k = 2$), so its cost here cannot be separated from noise; the random-mask result (68 extrapolated fills, 10%, erring at 6.6 pp MAE against 4.3 pp) is the one that measures it. The sparse runs show that for a sparsely measured model the pipeline fills 19–26% of the remaining cells, at an MAE within 0.8 pp of the random-mask result.

The pipeline left 266 of the 927 masked cells (29%) empty, because no mapping into that benchmark was available from the model's remaining measured scores, or the only eligible mappings had failed the quality gate or fallen, or the model's source score lay beyond the calibration's reach. On those cells the rank-2 completion erred at 11.8 pp MAE (16.9 RMSE, p90 30.3), double its 5.9 pp on the filled cells: the cells the pipeline leaves empty are those on which the completion has its largest errors.

On the identical 661 filled cells, the pipeline reaches 4.6 pp MAE against 5.9 pp for the rank-2 completion, which is 23% lower MAE (mean paired per-cell difference in absolute error, pipeline minus completion, -1.4 pp; -1.1 ± 0.5 pp across the three seeds), against 5.4 pp for the sweep's best rank (3; 16% lower), and against 12.2 pp for the per-benchmark mean; RMSE gives the same order (7.1 vs. 9.0 vs. 8.3 vs. 15.9). The completion's MAE is higher than the pipeline's at every confidence level (Table 4). The difference on the low-confidence cells is, however, small (7.5 vs. 8.9 pp MAE), so we tested the simplest hybrid, the mean of the pipeline's and the completion's per-cell predictions. Overall it matches the pipeline (4.6 pp MAE for both), but on the 109 low-confidence cells its MAE is lower than either component's (6.7 vs. 7.5 and 8.9 pp), so a hybrid that defers to the completion on flagged cells is worth exploring. The rank sweep changed the baseline little (ranks 2–4 all land at 7.1–7.6 pp MAE on all masked cells; rank 1 rises to 13.1). The completion fills 99% of the masked cells against the pipeline's 71%; its MAE is 5.9 pp on the cells the pipeline fills and 11.8 pp

Table 5 The largest harness-tax pairs. HLE: Humanity's Last Exam; τ-b: Kendall rank correlation between the two versions' scores; dir: share of models agreeing with the majority sign; p: two-sided sign test; candidate: auto-detected family without a tier.

benchmark	versions compared	tier	item set	n	mean $\lvert\Delta\rvert$	τ-b	dir	p
Automation Bench	AA vs published	candidate	weak	13	19.7	0.15	1.00	2×10^{-4}
Terminal-Bench 2.1	published vs VALS	agentic	verified	29	12.1	0.52	1.00	4×10^{-9}
HLE	tools off vs on	protocol	verified	11	10.0	0.81	1.00	10^{-3}
Terminal-Bench 2.1	VALS vs AA	agentic	verified	11	9.7	0.88	1.00	10^{-3}
HLE	published vs AA	protocol	needs audit	48	8.2	0.58	0.61	0.18
SWE-bench Verified	published vs VALS	agentic	needs audit	24	6.3	0.68	0.81	7×10^{-3}
IFBench	published vs AA	tool-free	verified	11	4.2	0.77	0.67	0.51
GPQA Diamond	published vs VALS	tool-free	verified	23	1.4	0.69	0.65	0.66

on those it leaves empty. These results are from one snapshot and a minimal baseline; BenchPress's full model adds constraints we have not implemented, so they do not show that factor models in general are less accurate.

79 of the 661 holdout fills (12%) came through multivariate mappings, and their lower in-sample error was not reproduced in MAE: 4.8 pp against 4.5 pp for the univariate fills (6.3 vs. 7.2 RMSE, a lower RMSE only, on 79 cells). The multivariate path is stored only when its cross-validated error is lower than the univariate one's; out of sample, the difference appeared only in RMSE, and we report the multivariate stage as unproven. On the univariate fills, estimates through mappings between versions from different sources (mixed-harness mappings) erred at 5.2 pp MAE against 4.2 pp through same-source ones ($n = 175$ vs. 407). This association is not a clean causal estimate: mixed-harness pairs are over-represented in agentic capabilities and in small-overlap pairs, both harder independently of harness.

5.4 Harness tax across benchmark families

Twenty-three families cover the tracked versions; 31 version pairs share at least one measured model, giving 378 model-level comparisons. Figure 5 ranks the pairs by mean $\lvert\Delta\rvert$ (pairs with fewer than five shared models are greyed out); Table 5 lists the largest.

Agentic benchmarks carry a systematic harness tax. On verified same-item-set pairs, agentic benchmark versions disagree about the same models by 8.7 pp on average (61 model-level comparisons over 4 pairs), against 2.3 pp for tool-free benchmarks (153 comparisons over 5 pairs), a ratio of 3.7. The largest verified agentic pair, Terminal-Bench 2.1 published vs. VALS, differs by 12.1 pp on average with a directionality of 1.00 ($p = 4 \times 10^{-9}$): the published numbers are systematically higher. The audit queue holds individual deltas up to 33 pp on this pair (DeepSeek V4 Pro 0813 scores 87.9% on the published leaderboard but 54.7% under VALS; Hy4 preview, 85.4% vs. 55.1%), which may reflect a scaffolding or retry-policy difference; item-level logs are not public, so the cause cannot be determined. Where a benchmark asks the model to act (navigate a terminal, drive tools), the harness's scaffolding, retry policy and scoring rules are part of what is being measured.

A protocol change alone can also shift scores by ten percentage points. Two pairs in the data differ in one protocol flag under the same source and share more than

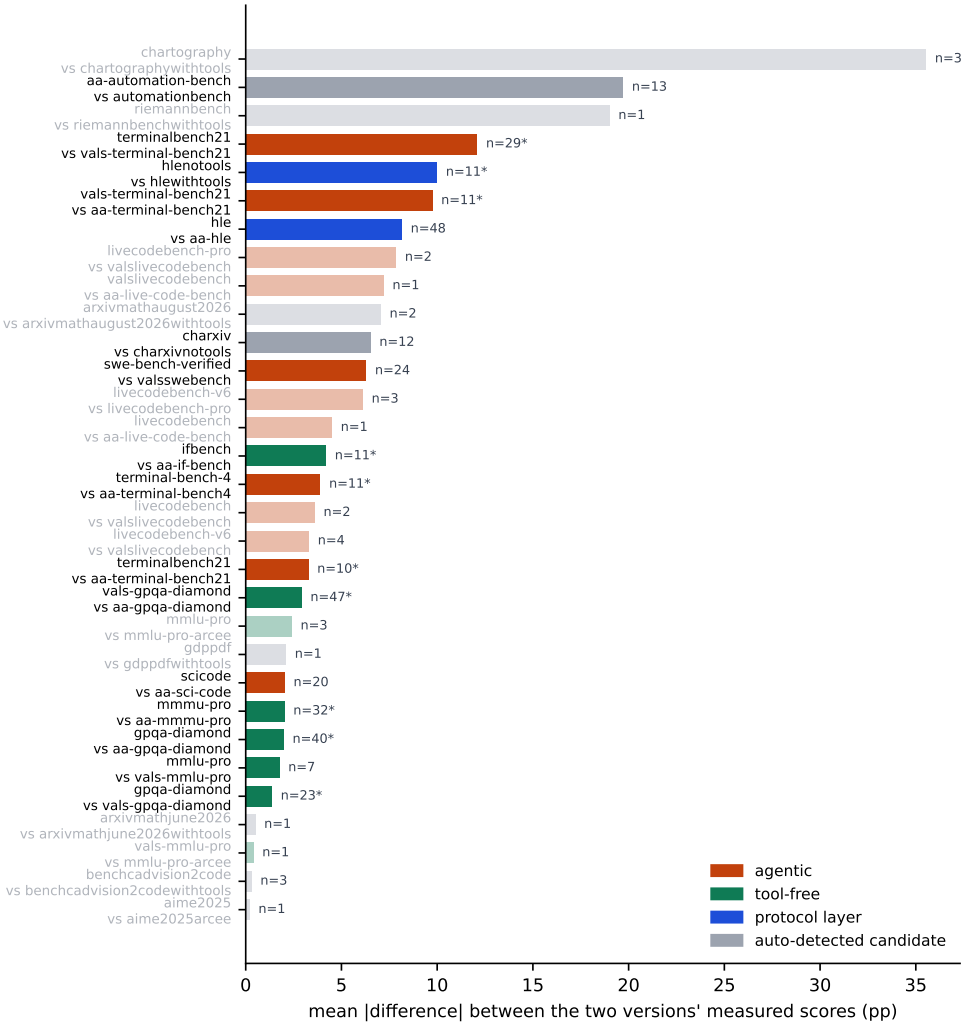


Fig. 5 The harness tax per benchmark-version pair. Verified same-item-set pairs are marked with an asterisk; pairs with fewer than five shared models are greyed out

three models: Humanity's Last Exam (HLE) and CharXiv, each against its no-tools variant. In both, the benchmark and the harness are the same and all models agree on the sign. All 11 models measured on both HLE versions score higher with tools, by 10.0 pp on average (Figure 6), sign test $p = 10^{-3}$, τ-b = 0.81. On CharXiv, an auto-detected family whose item alignment is not yet audited, all 12 models score higher than on the no-tools variant, by 6.5 pp on average (sign test $p = 5 \times 10^{-4}$, τ-b = 0.66). Six further tools on/off pairs share one to three models each, too few for a test, and the Terminal-Bench Science verifier variants share none. Two examples do not bound the effect on other benchmarks.

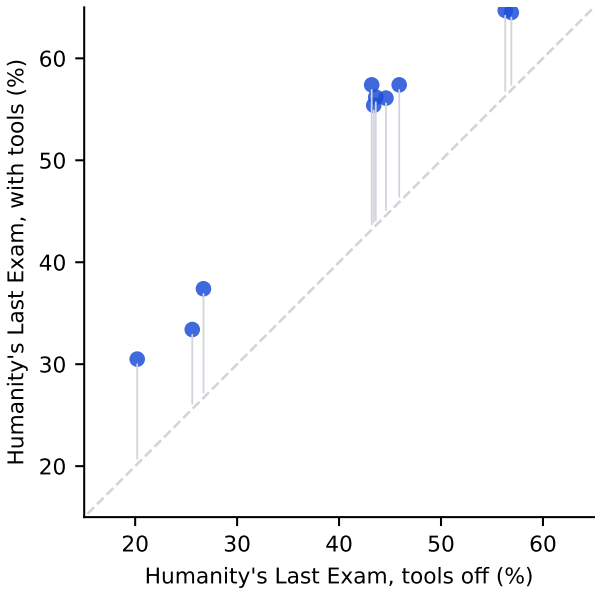


Fig. 6 Humanity's Last Exam with tools on versus off, per model

At the other end, tool-free knowledge benchmarks nearly agree across sources. GPQA Diamond (published vs. VALS, 1.4 pp, verified items) and one MMLU-Pro pair (published vs. VALS, $n = 7$, 1.8 pp; item set not verified, unlike the GPQA figure) disagree by less than 2 pp. Where there is no agent loop and no protocol freedom, harness choice matters least. IFBench (instruction following, no tools) is in the tool-free tier for the same reason, and its difference is statistically indistinguishable from zero (4.2 pp, $p = 0.51$).

The tier system encodes two qualifications. First, the HLE published vs. AA pair is tagged *protocol layer* in the registry because AA's run may include tools, but its alignment status is *needs audit*, and the data are consistent with that status: the mean difference is 8.2 pp but the directionality is weak (0.61, sign test $p = 0.18$), so no systematic claim is made. It is excluded from the verified-pairs headline. Second, one auto-detected family, Automation Bench (AA vs. published), shows the largest raw tax of all pairs with at least five shared models (19.7 pp, directionality 1.00, $p = 2 \times 10^{-4}$), but its alignment is weak, and part of the difference may come from different task sets rather than different harnesses. It stays without a tier until audited.

6 Limitations

The first limitation is shared by the whole research line: as Zhang et al. (2025) show, every score-prediction method misestimates models unlike its training set. benchgap's extrapolation flags, small-sample demotions and per-cell errors make the known risk factors visible, and the holdout evaluation shows that the confidence levels are correctly ordered, but nothing in the pipeline detects a model whose capability profile differs

from those in the fitting data. A model that scores high on the target but low on its source benchmarks can receive a large error with a high confidence level. The confidence levels flag the known risk factors, not unknown ones, and the holdout's masked cells are drawn from the same distribution as the fitting data, indeed from its easier end (Section 5.3 reweights accordingly).

The per-cell LOO error labels are optimistic: they are selected minima over families and mappings (Section 4.1), and in the holdout the realized RMSE is 1.4–1.7 times the label, depending on the level, with heavy tails. Nested cross-validation around the family selection, shrinking small samples' claimed errors toward a prior, or multiplying the published labels by these per-level factors would reduce the optimism; serving corrected labels through the API, with the selection bias noted in the error field's description, is planned. The rising-only rule applies to the stored fit; the leave-one-out refits that estimate its error are not constrained, so a fold may still fit a falling curve. Bounding every candidate's parameters so that it rises by construction would remove this inconsistency. Per-cell selection by lowest claimed error can also favour small-overlap mappings whose LOO error is small by chance; breaking ties toward larger overlap is a possible refinement.

Of the stored calibrations, 44% pair versions from different sources; the model-level part of the harness difference (up to tens of points on agentic benchmarks) enters their error, and the holdout shows higher error for such mappings (Section 5.3). A same-source constraint on calibration pairs would reduce coverage and remains to be tested.

The LOO error attached to each estimate is a point estimate of typical error, not a credible interval with correct coverage. A Bayesian version (a network over benchmark scores imputing every missing cell with one coherent posterior) is on the system's roadmap.

Finally, some choices are fixed rather than modelled. Capability gating is coarse: text benchmarks are never used for a vision benchmark, even where transfer would be safe. Successive releases of a benchmark are treated as unrelated targets. The quality-gate thresholds and the one-width reach are set by hand.

7 Discussion

The latent-factor line (Zeng and Papailiopoulos 2026; Maimon et al. 2025; Polo et al. 2024) shows that a few latent factors per model account for most of the score matrix. benchgap does not use that shared structure. On this snapshot, the explicit per-pair pipeline has a lower MAE than a matrix completion on the cells it fills (4.6 vs. 5.4 pp at the baseline's best rank), while filling 71% of the masked cells against the completion's 99%. The cells it leaves empty are those with no eligible within-capability mapping that passes the quality gate, rises and reaches the model's source score; on them the completion erred at 11.8 pp MAE. A global factor method is not restricted by capability or source, and its errors are reported in aggregate. In a public leaderboard each cell is read on its own, so benchgap publishes each estimate with its source benchmark, source score, curve and error.

The multivariate mappings are a deterministic precursor of a joint model: the elastic net's selected sources and the Michaelis–Menten weights show which benchmarks predict each other, which is the structure a Bayesian network over benchmark scores would encode. The roadmap is to move from point predictions with error bars to posterior imputation over the whole matrix while keeping constraints (i), (ii) and (v) of Section 4.3, which make the current estimates auditable.

A benchmark score is a property of the model under a specific evaluation setup, not of the model alone, yet compilations such as the one used here list scores from different sources side by side. On verified agentic pairs the mean disagreement between sources is 8.7 pp, and on Terminal-Bench 2.1 single models differ by up to 33 pp (Section 5.4). Any prediction method trained on scores from mixed sources inherits that disagreement as noise. benchgap records the source of every measured score and reports the per-pair disagreement; mappings between versions from different sources absorb the average difference, and the model-level remainder stays in their error.

8 Reproducibility

The pipeline is MIT-licensed and deterministic: it downloads the public BenchLM compilation, builds the SQLite database, fits every calibration from fixed data-driven starting values with selection by leave-one-out error, and refreshes daily via a scheduled workflow (benchgap contributors 2026). Most of what this paper reports is published through the free JSON/CSV API with an OpenAPI 3.1 description at https://benchgap.net/api: all scores (measured and estimated, with provenance and confidence), every univariate calibration with its parameters and metrics, and the harness-tax families, pairs and per-model deltas. The 21 multivariate mappings and the cross-domain fits are stored in the database and exported by the pipeline's scripts, but are not yet served by API v1, so they cannot be checked through the API alone. Because the site refreshes daily, a past snapshot cannot be regenerated from the live API; the snapshot analysed here is therefore archived on Zenodo with the harness-tax analysis, the holdout results and the scripts that produce every number and figure of this paper (Süli et al. 2026). Measured scores are data from BenchLM.ai, CC BY-NC 4.0; the estimates are benchgap's additions.

Acknowledgements. We gratefully acknowledge the support of the HUN-REN AI Service Center, which provides critical resources and infrastructure in the form of inspiration, education, consultation, and technology to the research community. Its mission to address researchers' needs in AI and to enable world-class results in scientific inquiry has been instrumental in advancing our work.

Statements and Declarations

Funding. Gy. Eigner was supported by the Distinguished Program of Obuda University.

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval and consent to participate. Not applicable. The study uses publicly reported benchmark scores of language models and involves no human participants or animals.

Data availability. The measured scores analysed in this study are from the public BenchLM.ai compilation, licensed CC BY-NC 4.0 (BenchLM.ai 2026). All measured and estimated scores of the current release, with provenance and confidence levels, every univariate calibration with its parameters and metrics, and the harness-tax families, pairs and per-model deltas are available through the free API at https://benchgap.net/api. The live data refreshes daily; the database snapshot of 8 October 2026 analysed here is archived on Zenodo at https://doi.org/10.5281/zenodo.23570459, together with the harness-tax analysis, the holdout results and the scripts that reproduce the numbers and figures of this paper (Süli et al. 2026).

Code availability. The pipeline is MIT-licensed and available at https://github.com/hollorol/benchgap (benchgap contributors 2026).

Author contributions. P.P. Süli and R. Hollós designed and built the benchgap system and website and wrote the manuscript together. Gy. Eigner supervised the project, provided the control perspective and secured funding. All authors reviewed and edited the manuscript.

References

benchgap contributors (2026) benchgap: an LLM benchmark leaderboard with estimated scores. https://benchgap.net, mIT-licensed pipeline, source at https://github.com/hollorol/benchgap. Accessed 8 October 2026

BenchLM.ai (2026) BenchLM.ai: public compilation of LLM evaluation results. https://benchlm.ai/data, dataset, CC BY-NC 4.0. Accessed 8 October 2026

Bowyer S, Locatelli A, Cao K (2026) Efficient benchmarking is just feature selection and multiple regression. arXiv preprint https://doi.org/10.48550/arXiv.2605.25773, arXiv:2605.25773

Kipnis A, Voudouris K, Schulze Buschoff LM, et al (2024) metabench: A sparse benchmark of reasoning and knowledge in large language models. arXiv preprint https://doi.org/10.48550/arXiv.2407.12844, arXiv:2407.12844

Maimon A, Cohen AD, Vishne G, et al (2025) From benchmarks to skills: Low-rank factors for LLM evaluation. arXiv preprint https://doi.org/10.48550/arXiv.2507.20208, retitled "IQ Test for LLMs: An Evaluation Framework for Uncovering Core Skills in LLMs" in v2, arXiv:2507.20208

Owen D (2024) How predictable is language model benchmark performance? arXiv preprint https://doi.org/10.48550/arXiv.2401.04757, arXiv:2401.04757

Pacchiardi L, Voudouris K, Slater B, et al (2025) PredictaBoard: Benchmarking LLM score predictability. arXiv preprint https://doi.org/10.48550/arXiv.2502.14445, arXiv:2502.14445

Park J, Mendes E, Stanovsky G, et al (2025) Look before you leap: Estimating LLM benchmark scores from descriptions. arXiv preprint https://doi.org/10.48550/arXiv.2509.20645, arXiv:2509.20645

Polo FM, Somerstep S, Choshen L, et al (2024) Sloth: scaling laws for LLM skills to predict multi-benchmark performance across families. arXiv preprint https://doi.org/10.48550/arXiv.2412.06540, arXiv:2412.06540

Ruan Y, Maddison CJ, Hashimoto T (2024) Observational scaling laws and the predictability of language model performance. arXiv preprint https://doi.org/10.48550/arXiv.2405.10938, arXiv:2405.10938

Süli PP, Eigner Gy, Hollós R (2026) benchgap: LLM benchmark score snapshot of 8 October 2026 and reproduction package for "Mind the Gap". Zenodo, https://doi.org/10.5281/zenodo.23570459, dataset, version 2026-10-08

Zeng Y, Papailiopoulos D (2026) You don't need to run every eval. arXiv preprint https://doi.org/10.48550/arXiv.2606.24020, arXiv:2606.24020

Zhang G, Dorner FE, Hardt M (2025) How benchmark prediction from fewer data misses the mark. arXiv preprint https://doi.org/10.48550/arXiv.2506.07673, arXiv:2506.07673