

BALANCE OF BENCHMARKS: SEMANTIC DENSITY REWEIGHTING FOR BENCHMARK MULTIPLICITY AND TASK-CONDITIONED EVALUATION

JHEN-KE LIN

National Yang Ming Chiao Tung University
jacob.cs14@nycu.edu.tw

ABSTRACT

Language models are commonly compared by averaging scores across a benchmark list with equal weight. Such lists grow through publication outside an explicit measurement design, so equal weighting turns the density of published benchmarks into an implicit capability weight: densely benchmarked regions count repeatedly. We introduce Balance of Benchmarks (BoB), which embeds benchmark descriptions and assigns each benchmark an inverse-density semantic weight. Nearby entries share aggregate influence at a disclosed density scale. After equating heterogeneous scores onto a common latent scale, a residual field uses the same geometry to condition model rankings on a task query. The two components serve distinct empirical roles. On a snapshot of 586 models and 14 benchmarks, BoB predicts which models are unusually strong on a held-out task beyond their general ability, reaching a profile correlation of 0.462 compared with 0.049 under equal weighting. It also limits the influence of densely repeated benchmarks on the aggregate. After adding four copies of each benchmark in turn, the resulting rankings retain Kendall $\tau = 0.995$, compared with 0.936 under equal weighting. The residual field therefore provides task-conditioned prediction, and inverse-density weighting provides robustness to benchmark multiplicity. Together, they turn benchmark-list composition from an incidental property of evaluation suites into an explicit, controllable part of measurement design, providing a principled foundation for task-aware and multiplicity-robust model evaluation.

1 INTRODUCTION

A leaderboard’s headline number is a weighted average, and its weights are usually equal. Equal weights appear neutral at the level of benchmark names, yet they imply a consequential choice at the level of capabilities: a region of capability space receives weight in proportion to the number of published benchmarks that happen to sample it. Benchmark lists grow by accretion, so publication density becomes capability weight. In the Artificial Analysis snapshot studied here, AIME and AIME’25 lie at angular distance 0.29 in an embedding of their descriptions, and the two Terminal-Bench variants at 0.36, against a median pairwise distance of 0.89 (Figure 1a). Competition mathematics and terminal agency therefore enter an equal-weight average more than once, with no explicit decision that they should matter more.

Existing aggregation strategies expose the same problem in different forms. Hand-assigned categories require discrete boundaries and give every benchmark inside a category the same relation to its neighbours. Weights estimated from cross-benchmark score covariance adapt to redundancy in the current model population, but the resulting scale changes when the population changes and can mistake common optimization or contamination for redundant coverage. Subset-selection methods ask which benchmarks add new information, but a leaderboard that retains the full list still needs continuous weights. The missing object is a model of how the benchmark list samples the task space.

Balance of Benchmarks (BoB) represents each benchmark by an embedded description and uses local description density to assign continuous weights over the observed list. Nearby entries share

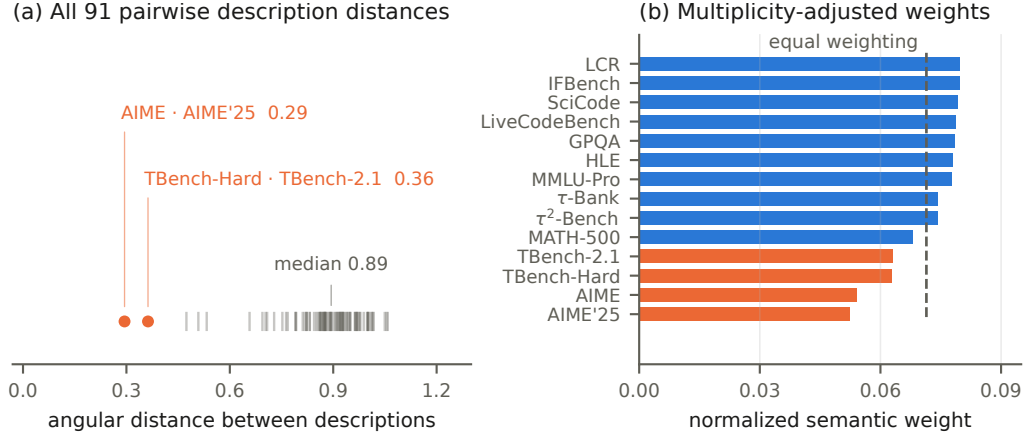


Figure 1: Publication density becomes aggregation weight. **(a)** All 91 pairwise angular distances among the 14 benchmark descriptions. The two closely related pairs highlighted in orange lie far below the median pair at 0.89. **(b)** Each benchmark’s description-derived semantic weight under BoB, against the $1/14$ nominal share of equal weighting.

aggregate influence at the chosen density scale, while isolated entries retain more. This construction corrects benchmark multiplicity without a capability taxonomy or duplicate threshold.

The same geometry also makes the aggregate task-conditioned. A user’s task is embedded as a query point, and a model’s global ability is adjusted by the residual performance it shows on nearby benchmarks. Heterogeneous benchmark scores must first be equated: 21% of models scored on AIME sit near its floor, while 20% of those scored on MATH-500 sit near its ceiling, so a linear standardization cannot align the ability ranges in which the two exams discriminate. Characteristic curves provide the common latent scale and the model-specific residuals used by the field (Section 3.1). The local adjustment is regularized by nearby observed support and returns to global ability as that support vanishes (Section 3.3).

We evaluate the construction by holding out each benchmark, fitting every rule on the other thirteen, and predicting the held-out benchmark’s actual scores. This tests whether the semantic space identifies which models are unusually strong for a task beyond general ability. A separate densification experiment measures what a benchmark adds as it moves through the space, while exact re-listing provides the limiting stress test. The distinction matters. Flattening the semantic weights leaves held-out profile prediction statistically comparable, but increases duplication-induced movement by roughly a factor of ten. Inverse-density weighting supplies robustness to benchmark multiplicity. The residual field supplies task-conditioned prediction.

Contributions. **(i) Benchmark multiplicity as an aggregation problem.** We show that equal weighting turns the number of published benchmarks per capability into an implicit capability weight and make benchmark-list composition an explicit part of aggregation design. **(ii) Continuous inverse-density reweighting.** Description-derived weights smoothly discount densely represented neighbourhoods with no category boundaries or duplicate threshold. The marginal-mass formula connects semantic densification to empirical robustness under re-listing. **(iii) Task-conditioned evaluation.** A query-conditioned residual field uses the same geometry to identify models that outperform their general ability near a task. On 586 models and 14 benchmarks, BoB reaches 0.902 level and 0.462 profile correlation in held-out prediction. Shuffled descriptions reach 0.336 profile on average and 0.430 at most, isolating the additional increment from correct semantic placement.

2 RELATED WORK

Leaderboard aggregation. Holistic suites and arenas made breadth a design goal while leaving the final aggregate to curation (Liang et al., 2023; Srivastava et al., 2023; Chiang et al., 2024; Singh et al., 2025; Artificial Analysis, 2026; Colombo et al., 2022). Under equal weighting, however, the

number of benchmarks assigned to a capability determines its contribution to the aggregate. Hand-maintained taxonomies make that choice explicit, but introduce category boundaries and maintenance costs (Hudson, 2026). BoB models the composition of the benchmark list directly, treating its entries as samples from a continuous task space.

Benchmark redundancy and selection. Score-side methods measure benchmark agreement, dimensionality, and predictability (Perlitz et al., 2024b; Ruan et al., 2024; Sha & Zhao, 2026). Related methods compress evaluation while preserving model rankings (Perlitz et al., 2024a; Maia Polo et al., 2024; Kipnis et al., 2025; Vivek et al., 2024). These estimates depend on the evaluated model population. A complementary line frames subset choice as marginal coverage, using description embeddings or score-side information (Yao et al., 2026; Smola, 2026). BoB carries the marginal-coverage view from subset choice to continuous weighting of the full list, with weights derived from descriptions and therefore exogenous to the population being ranked.

Semantic task representations. Representations of models and tasks support score prediction and model routing, including for unseen tasks (Zhuang et al., 2025; Song et al., 2025; Chen et al., 2025; Ge et al., 2026; Park et al., 2025; Li, 2025). BoB uses the same locality principle to return a task-conditioned ranking: nearby benchmark residuals adjust a global ability estimate. Because benchmark scores differ in difficulty and saturation, characteristic curves first place them on a common latent scale, following score-equating ideas from item response theory (Lalor et al., 2016; Rodriguez et al., 2021; Lord & Novick, 1968; Stocking & Lord, 1983; Kolen & Brennan, 2014). This calibration supplies comparable residuals. Inverse-density weighting and the query-conditioned residual field provide the aggregation mechanisms.

3 METHOD

BoB assumes that benchmark descriptions locate what their evaluations measure: benchmarks probing similar capabilities lie near one another in a semantic embedding, and a model’s performance varies smoothly over that space. Given a score matrix $S = (s_{mi})$ with missing entries and one description per benchmark, the method has three stages. Score equating places heterogeneous benchmarks on a common ability scale (Section 3.1). Inverse-density weighting adjusts each benchmark’s influence according to local description density (Section 3.2). A residual field conditions the aggregate on a task query (Section 3.3). We embed descriptions with Qwen3-Embedding-8B as unit vectors $x_i \in \mathbb{S}^{d-1}$, with $d(\cdot, \cdot)$ denoting angular distance. We use the encoder’s normalized outputs directly. Sections E and E.1 test the representation choices.

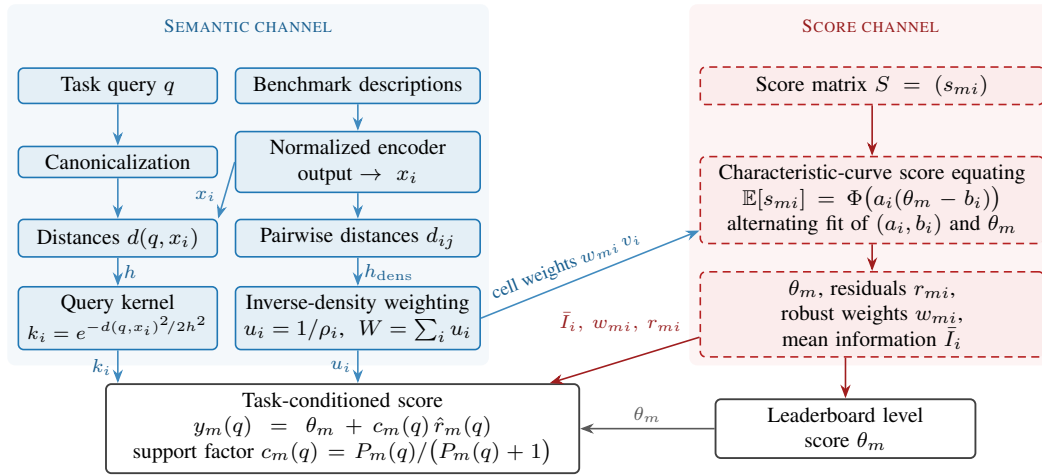


Figure 2: The two channels of BoB. Descriptions determine embeddings, distances, semantic weights, and the query kernel. Scores determine equated ability, standardized residuals, robust cell weights, and information. The semantic weights enter the ability fit, while no cross-benchmark score covariance enters the semantic channel.

3.1 SCORE EQUATING

A hard and an easy exam of the same trait discriminate in different ability ranges. In this snapshot, 21% of models scored on AIME lie within 0.05 of its floor, and 20% of those scored on MATH-500 lie within 0.05 of its ceiling (Figure 6a). A linear z -score shifts and rescales each column but cannot align these ranges. We therefore equate benchmark scores through a monotone characteristic curve (Lord & Novick, 1968; Embretson & Reise, 2000):

$$\mathbb{E}[s_{mi}] = \Phi(a_i(\theta_m - b_i)), \quad r_{mi} = \frac{s_{mi} - \Phi(a_i(\theta_m - b_i))}{\sigma_i}, \quad w_{mi} = \min(1, (3/|r_{mi}|)^2). \quad (1)$$

Here θ_m is model m 's latent ability, a_i and b_i are benchmark i 's discrimination and difficulty, Φ is the standard normal CDF, σ_i is its residual scale, and r_{mi} is the standardized residual. The robust cell weight w_{mi} limits the influence of large residual outliers in the public score matrix. Alternating estimation fits each (a_i, b_i) and then updates each ability by

$$\theta_m = \arg \min_{\theta} \sum_{i: s_{mi} \text{ observed}} w_{mi} v_i (s_{mi} - \Phi(a_i(\theta - b_i)))^2, \quad (2)$$

with the semantic weights v_i defined below. The level score is θ_m . The mean information $\bar{I}_i = M^{-1} \sum_m (a_i \phi(a_i(\theta_m - b_i)))^2 / \sigma_i^2$ records how strongly benchmark i separates the observed model population and is used only for precision weighting inside the residual field. Implementation details and a linear-scale comparison appear in Sections D and G.8.

3.2 SEMANTIC INVERSE-DENSITY WEIGHTING

The method defines a scale-dependent distribution over the observed benchmark list. Let

$$k(d) = \exp\left(-\frac{d^2}{2h_{\text{dens}}^2}\right), \quad \rho_i = \sum_j k(d_{ij}), \quad u_i = \frac{1}{\rho_i}, \quad W = \sum_i u_i. \quad (3)$$

ρ_i measures local benchmark density at x_i . Its inverse u_i gives benchmark i more influence when its semantic neighbourhood is sparsely represented. The normalized semantic weight is $v_i = u_i / \sum_j u_j$, and W is the list's effective benchmark mass at bandwidth h_{dens} . The optimization in Equation (2) uses the equivalent mean-one rescaling.

The effect of adding a benchmark is continuous in its location.

Proposition 1. *Add a benchmark at x with similarities $k_i = k(d(x, x_i))$ to an existing list. Its marginal effective mass is*

$$\Delta W = \frac{1}{1 + \sum_i k_i} - \sum_i \frac{k_i}{\rho_i(\rho_i + k_i)}.$$

It equals 1 in unoccupied space, decreases as similarity to existing points increases, and equals 0 when it coincides with a benchmark whose neighbourhood has no similarity to the rest of the list. Against one isolated benchmark at distance d , $\Delta W = (1 - k(d))/(1 + k(d))$.

Thus a benchmark in an unoccupied region contributes one unit, while a point in a dense region contributes progressively less. Co-located re-listings share approximately one benchmark's mass. The equality is exact for an isolated group and differs by the kernel mass that the group shares with its neighbours (Section B). The construction prices proximity directly and uses no category system or duplicate threshold. We set h_{dens} to 0.25 times the median pairwise distance. This design setting determines the semantic scale at which nearby entries share influence and gives $W = 12.5$ on the 14-benchmark snapshot. Across the sensitivity range in Section F, W falls from 14.0 at 0.05 times the median distance to 1.6 at the median distance.

3.3 TASK-CONDITIONED RESIDUAL FIELD

A user task embeds to a query point q . The task-conditioned score adjusts global ability by the model's residual performance on nearby benchmarks. With $k_i = \exp(-d(q, x_i)^2/2h^2)$, the raw

local residual, its observed support, and the final score are

$$\hat{r}_m^{\text{raw}}(q) = \frac{\sum_i k_i u_i \bar{I}_i w_{mi} r_{mi}}{\sum_i k_i u_i \bar{I}_i w_{mi}}, \quad P_m(q) = \sum_i k_i u_i w_{mi}, \quad y_m(q) = \theta_m + c_m(q) \hat{r}_m^{\text{raw}}(q), \quad (4)$$

where $c_m(q) = P_m(q)/(P_m(q)+1)$. The kernel k_i selects semantic neighbours, u_i prevents a dense cluster from dominating, $\bar{I}_i$ gives more precision to benchmarks that separate models, and w_{mi} limits outlying cells. $P_m(q)$ counts nearby observed support in benchmarks' worth. The coefficient $c_m(q)$ regularizes local adjustments when that support is sparse. At $P_m(q) = 0$, the adjustment is zero and the score returns to the global level θ_m . We use the fixed unit scale in $P/(P+1)$, which has no tuned parameter. The support-dependent form serves as a fixed regularization convention. A constant-shrinkage control gives comparable held-out accuracy (Table 2), locating predictive lift elsewhere.

Adding an average standardized residual to standardized ability is a scale convention. A first-order latent-offset alternative performs slightly worse on held-out profile, and information-weighting P_m changes neither conclusion (Section D.1). The field bandwidth h is selected from $\{0.3, 0.45, 0.6\}$ times the median distance by nested leave-one-benchmark-out validation. Queries are first rewritten into the descriptions' register, length, and facet structure. Section G.5 quantifies the effect of this canonicalization.

Illustrative query. Consider the query “Act as an autonomous software engineering agent that works in a terminal: plan multi-step changes, run build and test commands, edit files, and debug failures in a real repository without human intervention.” Canonicalization yields a description of autonomous software engineering in a real repository that names planning, editing, testing, debugging, tool use, code comprehension, and recovery across long sessions. Its nearest benchmarks are Terminal-Bench Hard at distance 0.701, Terminal-Bench 2.1 at 0.805, and LiveCodeBench at 0.876. Median support is 0.387. The largest absolute rank movers are devstral-2 from 381 to 265, gemini-3-5-flash-minimal from 179 to 69, and nova-2-0-omni-reasoning-medium from 215 to 321. The resulting top three are gpt-5-6-sol, claude-opus-5, and claude-opus-5-xhigh. This example illustrates the computation, while held-out benchmarks supply the quantitative validation.

3.4 DESCRIPTION-DERIVED SEMANTIC WEIGHTS

The semantic weights u_i and v_i are fixed by benchmark descriptions and h_{dens} . The score matrix subsequently supplies characteristic curves, model abilities, residual scales, robust cell weights, item information, and the nested selection of field bandwidth h . This separation keeps the target weighting independent of the model population while allowing the fitted ranking to use score-side evidence. Figure 2 summarizes the two-channel construction, and Section G.2 enumerates every use of scores within it. The design property is testable: Section 4.4 adds model families and measures how much each rule moves the models already listed.

4 EXPERIMENTS

4.1 HELD-OUT BENCHMARK PREDICTION

Protocol. We study the Artificial Analysis snapshot of 2026-08-29: 586 models with at least 5 of 14 benchmarks scored, spanning mathematics, knowledge, coding, agentic, instruction-following, and long-context tasks (Lightman et al., 2024; Rein et al., 2024; Phan et al., 2026; Wang et al., 2024; Jain et al., 2025; Tian et al., 2024; Pyatkin et al., 2025; Merrill et al., 2026; Yao et al., 2025; Barres et al., 2025). Per-benchmark cell coverage and the treatment of JSON nulls and observed zeros are reported in Section A.

We hold out one benchmark at a time, refit every rule on the remaining thirteen, and predict the held-out benchmark's actual scores. BoB receives the held-out description as a query point that was absent from the fit. We report two Spearman correlations across models. *Level* compares the predicted and observed rankings directly. *Profile* first removes a quadratic in pool ability from both rankings, testing whether a rule predicts which models are unusually strong on the held-out task beyond general ability. Means and standard errors are over 14 folds. Paired comparisons

Table 1: Leave-one-benchmark-out prediction of held-out benchmark scores on Artificial Analysis. Entries are mean Spearman correlations over 14 folds, with standard errors in parentheses; bold indicates the best evaluated method.

Prediction method	Level $\uparrow$	Profile $\uparrow$
BoB (ours)	0.902 (0.014)	0.462 (0.054)
Equal weight	0.877 (0.015)	0.049 (0.039)
Hand-assigned category	0.867 (0.017)	0.315 (0.064)
Nearest benchmark	0.772 (0.025)	0.135 (0.038)
AA Intelligence Index	0.881 (0.018)	0.333 (0.094)
<i>Methods fitted from the current score population</i>		
Score-covariance weights	0.657 (0.073)	0.092 (0.065)
First principal component	0.826 (0.021)	0.028 (0.077)

use one-sided Wilcoxon signed-rank tests with Holm correction over six baselines. Four use fixed rules—equal weighting, our hand-assigned categories, the nearest benchmark, and the published AA Intelligence Index—and two infer weights from the current score population, generalized least squares on score covariance and the first principal component (Section A.1). The field bandwidth is selected by nested validation. All folds choose the same value (Section F).

BoB attains the highest mean level and profile correlations among the evaluated methods (Table 1). Its profile advantage over equal weighting ($p < 0.001$, $p_{\text{Holm}} < 0.001$) and hand-assigned categories ($p = 0.010$, $p_{\text{Holm}} = 0.020$) survives correction. The two endogenous rules reach 0.092 and 0.028 profile, showing that covariance structure in the current model population predicts an unseen benchmark less accurately than the semantic construction.

Two controls refine what profile measures. Under cross-fitted isotonic removal of general ability, BoB reaches 0.446 profile, versus 0.012 for equal weighting, 0.302 for categories, and 0.349 for calibrated ability alone. The increment over calibrated ability is significant ($p = 0.0012$), so task conditioning adds signal beyond score equating (Table 7). Excluding the AIME and Terminal-Bench close-pair families gives 0.897 level and 0.409 profile. The advantages over equal weighting and nearest-benchmark prediction remain significant after Holm correction. The commercial index contains 10 of the 14 benchmarks. On the four folds outside the index, BoB reaches 0.935 level and 0.610 profile, versus 0.885 and 0.364 for the index, an illustrative leakage control reported in Table 5.

4.2 SEMANTIC DENSIFICATION AND DUPLICATION ROBUSTNESS

The marginal value of a benchmark falls smoothly as it enters an already sampled region (Figure 3a). An inserted point contributes 0.49 units at distance 0.33, 0.94 at distance 0.60, and 1.00 at distance 0.80, each within 0.02 of the isolated-pair formula. No duplicate threshold is involved. At the disclosed density bandwidth, the full list has effective mass $W = 12.5$. The listed AIME variants contribute leave-one-out masses of 0.38 and 0.30, and the two Terminal-Bench variants 0.57 and 0.58, compared with about 0.99 for long-context reasoning and IFBench (Figure 3b). After normalizing the final aggregate to equal-weight benchmark units, the AIME pair receives 1.49 units and the Terminal-Bench pair 1.76, compared with 2.00 for two isolated benchmarks.

Exact re-listing tests the limiting case. With four added copies, mean Kendall τ across the 14 re-listing scenarios is 0.995 for BoB and 0.936 for equal weighting. Averaged across scenarios, the within-scenario median and 95th-percentile BoB displacements are 0.5 and 4.3 ranks of 586. The top ten changes in only 2 of 14 scenarios, by one model each time, and the top 50 in one scenario. Equal weighting changes 1.6 top-ten models on average and as many as six, and 3.0 top-50 models on average and as many as eight. These statistics describe what duplication changes in the portion of the ranking a user sees. The scenario-averaged maximum displacement of 16.6 is reported in Section B.

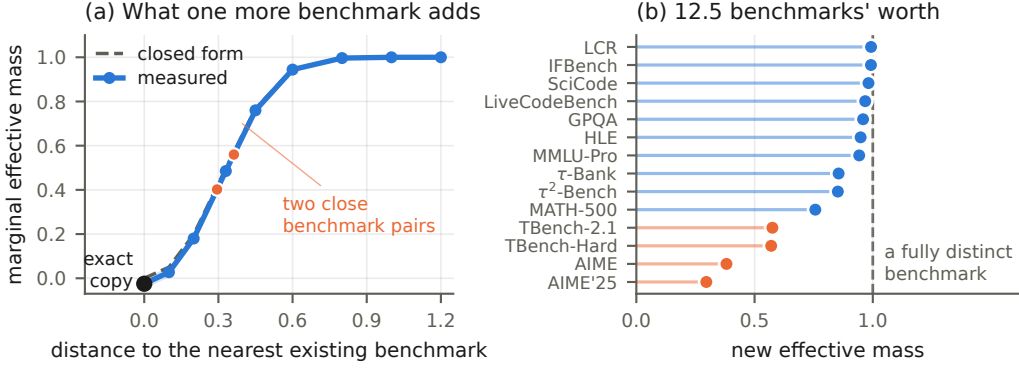


Figure 3: Semantic densification. **(a)** Marginal effective mass of an added benchmark against its distance to the nearest existing benchmark, using twelve random directions from each of the fourteen points (band: one standard deviation), with the isolated-pair closed form from Proposition 1. The orange markers locate the AIME and Terminal-Bench pairs. **(b)** Leave-one-out effective mass contributed by each listed benchmark.

4.3 ABLATIONS AND CONTROLS

Table 2: Component constructions and full-method controls on Artificial Analysis. Copy shift is the mean maximum displacement under four added copies, in ability standard deviations.

	Construction	Level $\uparrow$	Profile $\uparrow$	Copy shift (σ) $\downarrow$
A	Linear z , precision-weighted mean	0.889	0.014	0.763
B	Characteristic curves only	0.895	0.163	0.695
C	Residual field only, linear scale	0.896	0.089	0.343
D	Curves + field, no shrinkage	0.839	0.282	1.896
BoB	Curves + field + shrinkage (ours)	0.902	0.462	0.098
<i>Full-method controls</i>				
	No shrinkage	0.848	0.384	–
	Constant shrinkage	0.905	0.463	–
	Flat semantic weights ($u_i \equiv 1$)	0.896	0.451	0.973

The construction study locates the role of score equating. Characteristic curves alone reach 0.163 profile and a residual field on the linear scale 0.089, while the complete method reaches 0.462 (Table 2). These standalone constructions use their own ability controls, so the comparison establishes the ordering of the design choices across complete constructions. In the full method, removing shrinkage lowers level and profile to 0.848 and 0.384. A constant shrinkage matches BoB on profile (0.463) and is slightly better on level (0.905). Predictive lift is concentrated outside support dependence.

The two geometry controls separate prediction from multiplicity correction. Setting $u_i \equiv 1$ gives a statistically comparable profile of 0.451 while increasing duplication shift from 0.098 to 0.973 standard deviations. Inverse-density weighting therefore contributes primarily through robustness to benchmark multiplicity. Reassigning descriptions to benchmarks and rerunning the complete method 200 times gives mean profile 0.336 and maximum 0.430, below the observed 0.462 in every draw. Calibrated residual pooling helps under shuffled geometry. Correct semantic placement provides the remaining measurable increment. Paraphrased and lineage-stripped descriptions retain the result, and three embedding sizes move level and profile by at most 0.02 (Sections E and E.1).

Table 3: Population-shift test. Each row adds one model family. Entries are Spearman correlations between the pre-existing models’ rankings before and after the addition.

Family added	Models added	Score-covariance weights	BoB
Google	58	0.99519	0.99995
OpenAI	75	0.99300	0.99987
Anthropic	38	0.99862	0.99990

4.4 EMPIRICAL STABILITY UNDER POPULATION SHIFT

Adding 38–75 models from one family leaves the pre-existing BoB ranking at Spearman 0.9999 or higher (Table 3). Score-covariance weights move it to between 0.9930 and 0.9986. This experiment measures downstream population stability of the full fitted ranking. The description-derived semantic weights remain fixed while score-side quantities are refitted.

5 LIMITATIONS

Semantic proximity is an imperfect proxy for redundant coverage. A small distance can describe a re-listing, a fresh item sample from the same design, or an independent benchmark of the same capability. BoB assigns these cases the same geometric treatment, because using their score covariance would make the weight population-dependent. The representation can also miss a relationship. The τ^2 -Bench and τ -Bench Banking descriptions lie at distance 0.509, moving to 0.647 when their lineage sentence is removed. Their raw pair effective mass is 1.85. Global normalization gives them 2.08 equal-weight benchmark units, so the final aggregate does not discount the pair relative to equal weighting. A richer representation of capability overlap is the main route for improving this boundary.

The semantic distribution is representation dependent. The inverse-density weights define a distribution over one embedded point per benchmark at a chosen bandwidth. This benchmark-level representation leaves internal task extent and suite packaging outside the current estimator. Deployment settings can supply non-uniform target weights when particular capabilities have greater operational importance. All distances here come from one embedding family, and the evaluation contains 14 benchmarks. Three embedding sizes and paraphrased descriptions preserve the reported results within this representation family. The shuffled-description control bounds the semantic attribution. Generic residual pooling reaches 0.336 profile on average, while correct placement raises it to 0.462.

Re-listing robustness after refitting. The weighting result controls the mass assigned to co-located points, exactly for an isolated group and up to the similarity it shares with neighbours. Refitting characteristic curves and robust cell weights propagates the remaining change into model abilities. Across 14 four-copy scenarios, the average within-scenario median and 95th-percentile displacements are 0.5 and 4.3 ranks, and the mean maximum is 16.6. These are small relative to equal weighting but are not exact ranking invariance.

Held-out benchmarks are the easiest task-query regime. The task-conditioned rankings have no direct ground truth for arbitrary user tasks, so we evaluate them by querying a held-out benchmark description. Such a query is drawn from the same construction process as the remaining benchmark descriptions and is typically closer to the observed list than an unconstrained user request. External task distributions and prospective model selection are needed to test the method in deployment.

6 CONCLUSION

Equal weighting makes the density of published benchmarks part of the measurement, even when that density was never chosen. BoB makes the choice explicit: score equating places heterogeneous benchmarks on a common scale, inverse-density weighting adjusts the influence of densely represented semantic neighbourhoods, and the same task space conditions an aggregate on a user’s query.

The held-out prediction and shuffled-geometry results support task conditioning, while the densification and duplication experiments identify the distinct role of inverse-density weighting. The residual field supports task-conditioned prediction beyond calibrated general ability. Inverse-density weighting stabilizes the aggregate under benchmark multiplicity. Together they expose benchmark-list composition as an explicit measurement decision.

REPRODUCIBILITY STATEMENT

The Artificial Analysis snapshot is fixed at 2026-08-29 and restricted to the 586 models with at least 5 of the 14 benchmarks scored. Every experiment writes a frozen result file. One script regenerates every table from those files and a second regenerates every figure. The accompanying artifact contains the anonymized code, frozen results, derived benchmark descriptions, raw task queries, canonicalized queries, embeddings, fitted curve parameters, and both regeneration scripts. Section G.6 lists every query pair used in the qualitative examples. No model training is involved. Compute consists of embedding inference over short texts followed by CPU fitting of score equating, inverse-density weighting, and the residual field.

THE USE OF LARGE LANGUAGE MODELS

Large language models enter this work in three distinct roles. They are the object of study, since the systems in the evaluation snapshot are language models. They are also a component of the method. The English benchmark descriptions and their paraphrases are generated by a language model from a fixed template. The five demonstration queries are canonicalized into the same descriptive register. These frozen text inputs and the embedding configuration are specified in Section G.3, and canonicalization is quantified in Section G.5. Finally, language models were used as a writing and coding assistant, for drafting and editing prose and for implementation support, with the authors verifying the output. Every reported number is produced by the released scripts from the frozen result files rather than by a language model. The generated descriptions are not neutral inputs, however. They are the object the geometry is computed on, and four of them originally named their relationship to another benchmark, which Section E tests by removing every such reference and refitting.

REFERENCES

- Artificial Analysis. Artificial analysis: Independent analysis of AI models and API providers. <https://artificialanalysis.ai/>, 2026. Accessed 2026-08-29.
- Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2 -Bench: Evaluating conversational agents in a dual-control environment. arXiv:2506.07982, 2025. Preprint; no peer-reviewed venue as of 2026-08.
- Jianhao Chen, Chenxu Wang, Gengrui Zhang, Peng Ye, Lei Bai, Wei Hu, Yuzhong Qu, and Shuyue Hu. Learning compact representations of LLM abilities via item response theory. arXiv:2510.00844, 2025. Preprint.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388, 2024.
- Pierre Colombo, Nathan Noiry, Ekhine Irurozki, and Stephan Cl  men  on. What are the best systems? new perspectives on NLP benchmarking. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022.
- Susan E. Embretson and Steven P. Reise. *Item Response Theory for Psychologists*. Lawrence Erlbaum Associates, Mahwah, NJ, 2000.
- Chris Ge, Daria Kryvosheieva, Daniel Fried, Uzay Girit, and Kaivalya Hariharan. Agent psychometrics: Task-level performance prediction in agentic coding benchmarks. arXiv:2604.00594, 2026. Preprint.

- Blair Hudson. Meta-benchmarks for financial-services LLM evaluation. arXiv:2607.01740, 2026. Preprint.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025.
- 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 (ICLR)*, 2025.
- Michael J. Kolen and Robert L. Brennan. *Test Equating, Scaling, and Linking: Methods and Practices*. Springer, New York, NY, 3rd edition, 2014.
- John P. Lalor, Hao Wu, and Hong Yu. Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 648–657, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1062.
- Yang Li. Rethinking predictive modeling for LLM routing: When simple kNN beats complex learned routers. arXiv:2505.12601, 2025. Preprint.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024. Introduces the MATH-500 subset of the MATH test set.
- Frederic M. Lord and Melvin R. Novick. *Statistical Theories of Mental Test Scores*. Addison-Wesley, Reading, MA, 1968.
- Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326, 2024.
- Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, et al. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. arXiv:2601.11868, 2026. Preprint; introduces Terminal-Bench 2.0.
- Jungsoo Park, Ethan Mendes, Gabriel Stanovsky, and Alan Ritter. Anticipatory evaluation of language models. arXiv:2509.20645, 2025. Preprint; introduces the PRECOG corpus.
- Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. Efficient benchmarking (of language models). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2519–2536. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.naacl-long.139.
- Yotam Perlitz, Ariel Gera, Ofir Arviv, Asaf Yehudai, Elron Bandel, Eyal Shnarch, Michal Shmueli-Scheuer, and Leshem Choshen. Do these LLM benchmarks agree? fixing benchmark evaluation with BenchBench. arXiv:2407.13696, 2024b. Preprint.

- Long Phan, Alice Gatti, Nathaniel Li, Adam Khoja, Ryan Kim, et al. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649(8099):1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. Introduces the Humanity’s Last Exam (HLE) benchmark; preprint arXiv:2501.14249.
- Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems 38 (NeurIPS), Datasets and Benchmarks Track*, 2025. Introduces the IFBench benchmark; preprint arXiv:2507.02833.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *Conference on Language Modeling (COLM)*, 2024.
- Pedro Rodriguez, Joe Barrow, Alexander Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4486–4503. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.346.
- 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 37 (NeurIPS)*, 2024.
- Tommy Sha and Stella Zhao. BenchScope: How many independent signals does your benchmark provide? arXiv:2603.29357, 2026. Preprint.
- Shivalika Singh, Yiyang Nan, Alex Wang, Daniel D’Souza, Sayash Kapoor, Ahmet Üstün, Sanmi Koyejo, Yuntian Deng, Shayne Longpre, Noah A. Smith, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. The leaderboard illusion. arXiv:2504.20879, 2025. Preprint.
- Alexander Smola. Submodular benchmark selection. arXiv:2605.02209, 2026. Preprint.
- Wei Song, Zhenya Huang, Cheng Cheng, Weibo Gao, Bihan Xu, GuanHao Zhao, Fei Wang, and Runze Wu. IRT-Router: Effective and interpretable multi-LLM routing via item response theory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15629–15644. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.761.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.
- Martha L. Stocking and Frederic M. Lord. Developing a common metric in item response theory. *Applied Psychological Measurement*, 7(2):201–210, 1983. doi: 10.1177/014662168300700208.
- Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. SciCode: A research coding benchmark curated by scientists. In *Advances in Neural Information Processing Systems 37 (NeurIPS), Datasets and Benchmarks Track*, 2024.
- Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1601, St. Julian’s, Malta, 2024. Association for Computational Linguistics.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang,

- Rongqi Fan, Xiang Yue, and Wenhua Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 37 (NeurIPS), Datasets and Benchmarks Track*, 2024.
- Jihan Yao, Gantavya Bhatt, Arnav Das, Peter Jin, Ke Bao, Qiaolin Yu, Khushi Bhardwaj, Chang Su, Jialei Wang, Yikai Zhu, Sugam Devare, Damon Mosk-Aoyama, Zhen Dong, Venkat Krishna Srinivasan, Yineng Zhang, Oleksii Kuchaiev, Jiantao Jiao, Banghua Zhu, and Jeff Bilmes. Core-sets before score sets: Evaluation-unsupervised prompt subset selection for LLM benchmarks. arXiv:2607.09739, 2026. Preprint.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations (ICLR)*, 2025.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. arXiv:2506.05176, 2025. Technical report.
- Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning compact representations of large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

A DATA AND BASELINES

Artificial Analysis snapshot. We retrieved the snapshot directly from the Artificial Analysis Data API at <https://artificialanalysis.ai/api/v2/data/llms/models> at 2026-08-29T08:02:03Z. The saved JSON contains 624 API rows and has SHA-256 digest 0b8dee14982f1c244549ba0f17bd91e86b02034adf867a63cad431b68f7c88ed. We use AIME, AIME’25, MATH-500, GPQA Diamond, HLE, MMLU-Pro, LiveCodeBench, SciCode, IFBench, long-context reasoning, Terminal-Bench Hard, Terminal-Bench 2.1, τ^2 -Bench, and τ -Bench Banking. A row enters the analysis when at least five of these fourteen evaluation fields are numeric, leaving 586 rows. Each API row is one observation, so separately listed model configurations remain separate observations. The population-shift analysis groups rows by the API’s `model_creator.name` field.

A JSON `null` is represented as a missing cell and every numeric value, including zero, is retained as observed. The curve fit and residual field sum over observed cells. Robust cell weights attenuate large standardized residuals during fitting. Table 4 reports the resulting coverage.

Table 4: Cell coverage among the 586 retained Artificial Analysis API rows. Observed zero is a subset of observed. Null counts are values represented by JSON `null` in the saved snapshot.

Benchmark	Observed	Null	Observed zero
AIME	192	394	14
AIME’25	269	317	4
GPQA Diamond	585	1	0
HLE	583	3	0
IFBench	449	137	0
Long-context reasoning	516	70	64
LiveCodeBench	343	243	0
MATH-500	191	395	0
MMLU-Pro	345	241	0
SciCode	583	3	3
τ^2 -Bench	439	147	23
τ -Bench Banking	184	402	0
Terminal-Bench Hard	432	154	48
Terminal-Bench 2.1	216	370	5

A.1 BASELINE SPECIFICATION

Equal weighting averages the standardized pool scores. *Hand-assigned categories* averages the pool benchmarks sharing the held-out benchmark’s label. The snapshot provides no taxonomy, so the labels are ours: mathematics (AIME, AIME’25, MATH-500), knowledge (GPQA Diamond, HLE, MMLU-Pro), coding (LiveCodeBench, SciCode), agentic (τ^2 -Bench, τ -Bench Banking, and both Terminal-Bench variants), instruction following (IFBench), and long context. The last two categories contain one benchmark each. When that benchmark is held out, the category rule falls back to equal weighting. *Nearest benchmark* returns the standardized score of the closest observed pool benchmark in description space.

The *Artificial Analysis Intelligence Index* is used exactly as published. It is a composite of ten of the fourteen benchmarks, but the snapshot does not document its normalization. Among the 248 models with all ten components, the published index correlates at Spearman 0.938 with their equal-weight raw mean, so we treat it as an opaque baseline.

Two evaluation-side oracles estimate structure from the current score population. *Score-covariance GLS* uses $w \propto \Sigma^{-1}\mathbf{1}$, where Σ is the empirical correlation matrix of row-centred standardized pool scores with ridge 0.05. *First principal component* takes the leading component of the standardized pool matrix after filling missing cells with column means and aligns its sign with the mean score. Neither oracle enters BoB.

B INVERSE-DENSITY WEIGHTING AND DUPLICATION DECOMPOSITION

Write $\rho_i = \sum_j k(d_{ij})$ for the density at x_i before adding a new benchmark at x , and let $k_i = k(d(x, x_i))$. The new point raises each existing density to $\rho_i + k_i$ and has density $\rho_x = 1 + \sum_i k_i$. Therefore

$$\Delta W = \frac{1}{1 + \sum_i k_i} + \sum_i \left(\frac{1}{\rho_i + k_i} - \frac{1}{\rho_i} \right) = \frac{1}{1 + \sum_i k_i} - \sum_i \frac{k_i}{\rho_i(\rho_i + k_i)}.$$

Each term decreases componentwise as the similarities k_i increase. If every $k_i = 0$, then $\Delta W = 1$. If x coincides with a member of an isolated co-located group G of size n_G , then $k_i = 1$ and $\rho_i = n_G$ for $i \in G$, giving

$$\Delta W = \frac{1}{1 + n_G} - \frac{n_G}{n_G(n_G + 1)} = 0.$$

Against one isolated benchmark at distance d , $\rho_1 = 1$ and $k_1 = k(d)$ yield $\Delta W = (1 - k)/(1 + k)$.

The zero is exact only for an isolated group. If a co-located benchmark has neighbours, its copy also increases their densities and ΔW may be slightly negative. Equivalently, a group of n_G co-located benchmarks holds unnormalized mass

$$\frac{n_G}{n_G + c_G}, \quad c_G = \sum_{j \notin G} k(d(x_G, x_j)).$$

This differs from the single-copy value $1/(1 + c_G)$ by at most c_G . For benchmarks outside a close pair, c_G ranges from 0.003 to 0.16 on this snapshot. It is 0.46–0.50 for the AIME pair.

Why the fitted ranking still moves. We separate three sources under four added copies. Forcing the duplicated group’s weight to be exactly invariant and holding item parameters fixed moves no model. Restoring the actual inverse-density weights while still holding item parameters fixed produces a scenario-averaged maximum displacement of 10.6 ranks from the shared-neighbour $O(c_G)$ term. Refitting characteristic curves and robust cell weights raises the scenario-averaged maximum to 16.6. The corresponding maximum change in θ is 0.098 standard deviations, while adjacent models are a median 0.0045 apart.

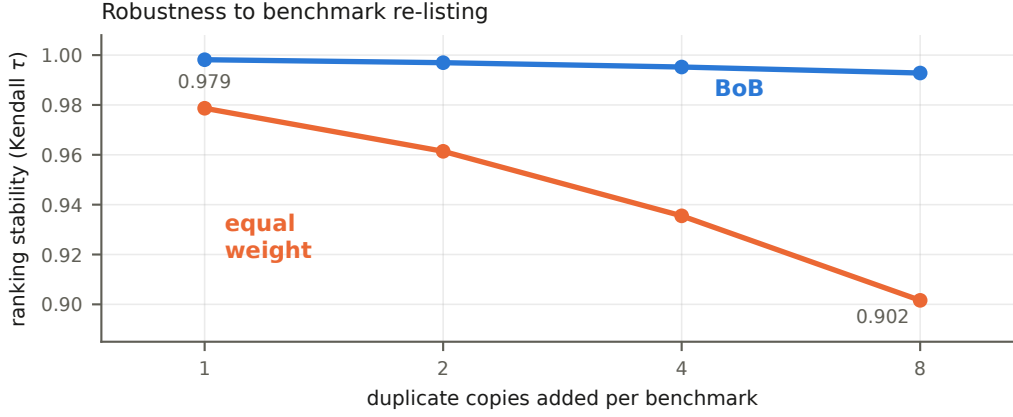


Figure 4: Ranking stability as each benchmark is re-listed with 1, 2, 4, or 8 additional identical score columns. The plotted value is the mean Kendall τ between the original and perturbed rankings over the fourteen re-listing scenarios.

C HELD-OUT EVALUATION DETAILS

Table 5: Held-out performance split by whether the benchmark is included in the AA Intelligence Index. Bold indicates the best entry in each column.

Prediction method	Held-out benchmark inside the index (10 folds)		outside it (4 folds)	
	Level $\uparrow$	Profile $\uparrow$	Level $\uparrow$	Profile $\uparrow$
BoB (ours)	0.888	0.403	0.935	0.610
Equal weight	0.874	0.071	0.885	-0.007
Hand-assigned category	0.858	0.282	0.888	0.397
Nearest benchmark	0.772	0.131	0.774	0.145
AA Intelligence Index	0.879	0.320	0.885	0.364

What level and profile measure. A quadratic in pool ability explains a mean $R^2 = 0.785$ of the held-out benchmark ordering, ranging from 0.55 to 0.92 across folds. A rule that reports pool ability alone therefore reaches 0.883 level. It reaches 0.094 on the quadratic profile metric because subtracting a quadratic does not remove every monotone transformation of ability. The stricter isotonic profile below closes that route. The 10-fold close-pair sensitivity excludes AIME/AIME’25 and Terminal-Bench Hard/2.1.

Table 6: Paired profile comparisons between BoB and each baseline. Entries are mean per-fold differences, one-sided Wilcoxon signed-rank p -values, and Holm-adjusted p -values.

BoB minus	All 14 folds			10 folds excluding the two primary close-pair families		
	Δ profile	p	p_{Holm}	Δ profile	p	p_{Holm}
Equal weight	+0.414	< 0.001	< 0.001	+0.341	0.002	0.012
Hand-assigned category	+0.147	0.010	0.020	+0.152	0.032	0.073
Nearest benchmark	+0.327	< 0.001	< 0.001	+0.303	0.002	0.012
AA Intelligence Index	+0.129	0.292	0.292	+0.070	0.423	0.423
Score-covariance weights	+0.370	0.003	0.010	+0.258	0.024	0.073
First principal component	+0.434	0.002	0.006	+0.343	0.014	0.055

For Table 7, the binned control is the sample-size-weighted mean Spearman correlation within ability deciles. Bins with fewer than 15 models or a degenerate variable are omitted.

Table 7: Profile correlation after three controls for general ability. The final column reports one-sided Wilcoxon tests against BoB under isotonic residualization.

Prediction method	Isotonic	Binned	Quadratic	p (isotonic)
BoB (ours)	0.446	0.486	0.462	–
Equal weight	0.012	0.210	0.049	< 0.001
Hand-assigned category	0.302	0.315	0.315	0.015
Nearest benchmark	0.136	0.214	0.135	< 0.001
AA Intelligence Index	0.317	0.372	0.333	0.292
Pool ability alone	−0.021	0.243	0.094	< 0.001
Calibrated θ alone	0.349	0.411	0.368	< 0.001

Under isotonic residualization, calibrated θ alone reaches 0.349 and BoB reaches 0.446. Their paired difference is significant ($p = 0.0012$), locating the task-conditioned increment beyond score equating. The paired tests over all 14 folds and the primary close-pair sensitivity are reported in Table 6. In the 10-fold sensitivity, only equal weighting and nearest benchmark survive Holm correction.

Table 8: Per-fold leave-one-benchmark-out results for BoB and the four fixed-rule baselines included in the primary comparison.

Benchmark	BoB		Equal weight		Category		Nearest		AA index	
	Level	Profile	Level	Profile	Level	Profile	Level	Profile	Level	Profile
AIME	0.957	0.778	0.856	-0.305	0.907	0.528	0.789	0.189	0.799	-0.062
AIME'25	0.875	0.631	0.833	-0.084	0.914	0.740	0.606	0.061	0.745	-0.289
GPQA Diamond	0.964	0.603	0.955	0.093	0.947	0.464	0.875	0.187	0.947	0.384
HLE	0.904	0.614	0.863	0.198	0.880	0.431	0.839	0.143	0.859	0.264
IFBench	0.836	0.333	0.808	0.111	0.808	0.111	0.721	0.229	0.823	0.277
Long-context reasoning	0.895	0.278	0.894	0.042	0.894	0.042	0.877	0.281	0.929	0.528
LiveCodeBench	0.936	0.555	0.898	-0.115	0.830	-0.030	0.834	-0.025	0.849	-0.040
MATH-500	0.955	0.686	0.897	-0.027	0.927	0.590	0.802	0.246	0.824	-0.074
MMLU-Pro	0.915	0.201	0.950	0.159	0.959	0.524	0.686	-0.012	0.917	0.287
SciCode	0.896	0.105	0.922	0.145	0.830	-0.015	0.846	0.035	0.927	0.412
τ^2 -Bench	0.780	0.362	0.738	0.042	0.741	0.268	0.665	0.153	0.846	0.627
τ -Bench Banking	0.887	0.352	0.867	0.026	0.836	0.186	0.630	-0.176	0.937	0.719
Terminal-Bench Hard	0.880	0.347	0.877	0.117	0.782	0.284	0.767	0.256	0.950	0.753
Terminal-Bench 2.1	0.943	0.625	0.921	0.278	0.882	0.286	0.875	0.322	0.979	0.875
Mean	0.902	0.462	0.877	0.049	0.867	0.315	0.772	0.135	0.881	0.333

D ABLATIONS AND VALIDITY CHECKS

Rows A–D of Table 2 are self-contained constructions. They use their construction-specific ability control, whereas the full-method controls use the pool-only evaluation path of the headline experiment. The rows order design choices but do not form an additive decomposition. The additional linear z -score and precision-weighted-mean construction reaches 0.014 profile.

D.1 GEOMETRY CONTROLS

Flat weighting. Setting $u_i \equiv 1$ throughout the otherwise unchanged method gives 0.896 level, 0.451 profile, and a duplication shift of 0.973 standard deviations, versus 0.098 for BoB. Its profile is statistically comparable to BoB ($p = 0.077$). Inverse-density weighting contributes multiplicity robustness.

Shuffled descriptions. Reassigning descriptions to benchmarks and rerunning the complete leave-one-benchmark-out procedure 200 times yields profile 0.336 ± 0.042 (maximum 0.430), versus the observed 0.462. This separates the contribution of correct semantic placement from generic pooling of calibrated residuals.

Residual-to-ability scale. Equation (4) adds an average standardized residual to standardized ability. A first-order latent-offset alternative divides each cell residual by the local derivative of its characteristic curve and weights by local information. It gives 0.904 level and 0.435 profile, providing a sensitivity check on the reported direct standardized-residual convention.

Support-factor specification. Including normalized item information inside $P_m(q)$ gives 0.905 level and 0.460 profile, indistinguishable on profile ($p = 0.67$). The conclusion is unchanged when information enters both the residual average and its support factor.

E TEXT AND REPRESENTATION ROBUSTNESS

Paraphrased descriptions. Replacing every benchmark description with an independently generated paraphrase gives 0.903 level and 0.474 profile, preserving the held-out result.

Lineage-stripped descriptions. Four generated descriptions explicitly name a relationship to another benchmark. We remove every cross-benchmark reference, re-embed, and rerun the analysis. AIME and AIME'25 move from distance 0.295 to 0.227 and remain the closest pair. The Terminal-Bench variants move from 0.363 to 0.461 and remain second. Held-out prediction remains 0.900 level and 0.454 profile, compared with 0.902 and 0.462. The τ pair moves from 0.509 to 0.647, which exposes the representation false negative discussed in Section 5.

E.1 EMBEDDING SIZE

Table 9: Primary quantities recomputed with three sizes of the Qwen3 embedding family (Zhang et al., 2025). The main text uses the 8B column.

	Qwen3-Emb 0.6B	Qwen3-Emb 4B	Qwen3-Emb 8B
AA level	0.902	0.902	0.902
AA profile	0.464	0.462	0.462
AA copy shift in ranks (equal weight 96.9)	24.4	16.1	16.6

Level and profile vary by less than 0.01 across sizes. Four-copy displacement is 24.4 ranks for 0.6B, 16.1 for 4B, and 16.6 for 8B, below equal weighting’s 96.9.

F SENSITIVITY AND SELECTION

Field bandwidth. The field bandwidth is selected over $\{0.3, 0.45, 0.6\}$ times the median pairwise distance, using the mean of level and profile under leave-one-benchmark-out validation. The criterion is 0.655, 0.682, and 0.660, selecting $h = 0.402$. In a nested variant, each fold selects h using the other thirteen folds. Every fold selects 0.45 times the median, and the nested result is the reported 0.902 level and 0.462 profile.

Density bandwidth. The rule $h_{\text{dens}} = 0.25$ times the median distance gives $h_{\text{dens}} = 0.224$ and $W = 12.5$. Across Figure 5a, W falls from 14.0 at 0.05 times the median distance to 1.6 at the median itself. Effective benchmark mass is indexed by this explicit scale.

Robust constants and noise floors. Sweeping the robust cap over $\{2, 3, 4, 6\}$ and its floor over $\{10^{-2}, 10^{-3}, 10^{-4}\}$ moves level by at most 0.004 and profile by at most 0.008. The floor is inert over this range. On Artificial Analysis, per-benchmark residual variances use binomial sampling floors because per-question scores are unavailable. Doubling or halving these floors leaves the benchmark weights at Spearman 0.991 and 0.987, and dropping GPQA Diamond leaves the fitted abilities at Spearman 0.998.

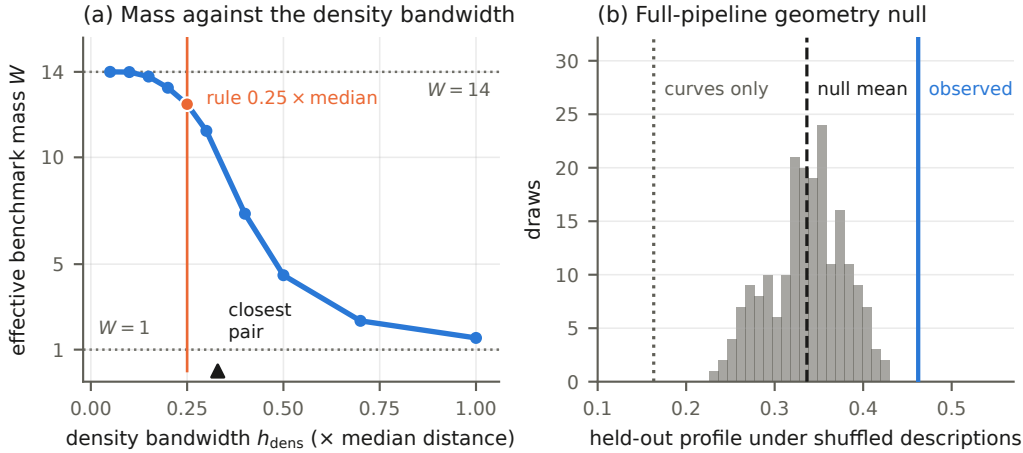


Figure 5: **(a)** Effective benchmark mass W against the density bandwidth, with the 0.25-times-median rule and the closest observed pair marked. **(b)** Complete-method profile under 200 shuffled-description runs. The null mean is 0.336 and no run reaches the observed 0.462.

G SEMANTIC AND SCORE CHANNELS

The construction maintains separate semantic and score channels. Figure 2 gives the complete flow in the main method. Figure 6 illustrates score equating across difficulty ranges, Table 10 enumerates the score-side inputs, and Figure 7 expands the factors in the task-conditioned field.

G.1 SCORE EQUATING ACROSS DIFFICULTY RANGES

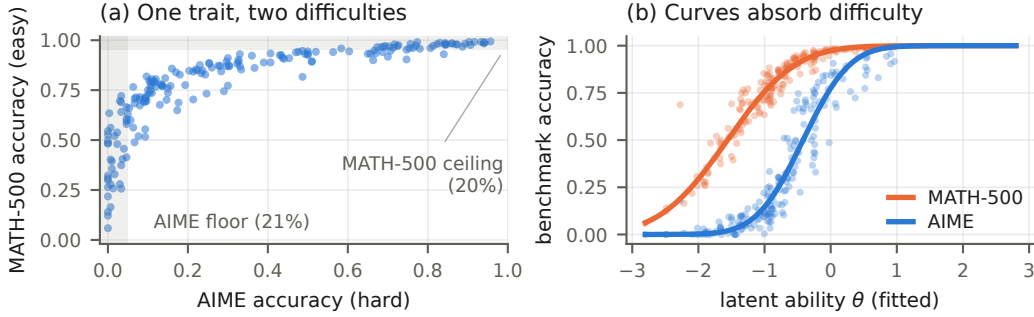


Figure 6: Why scores require equating. **(a)** AIME and MATH-500 test related mathematics at different difficulties: many models lie at the AIME floor or the MATH-500 ceiling. **(b)** Their fitted characteristic curves align the ranges in which each benchmark discriminates.

G.2 WHERE SCORES TOUCH THE CONSTRUCTION

Table 10: Uses of model scores in the primary construction. The semantic weights are fixed from descriptions before these score-side quantities enter.

Use of scores	Enters as	Role
Characteristic-curve fit	a_i, b_i, θ_m in Equations (1) and (2)	equates scales and supplies residuals
Per-benchmark residual scale	σ_i in r_{mi} and $\bar{I}_i$	diagonal scale estimate with binomial floor
Mean item information	$\bar{I}_i$ in Equation (4)	precision-weights the local residual average
Population location and scale	standardization of s_{mi} and θ_m	sets units, not semantic weight direction
Robust cell weight	w_{mi} in Equations (2) and (4)	limits the influence of one cell’s residual
Cross-validation of h	field smoothness	evaluation-side selection with a nested check

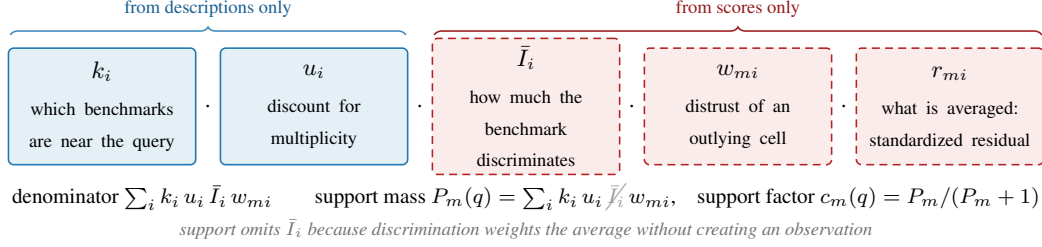


Figure 7: Factors in the numerator of Equation (4). The query kernel and inverse density come from descriptions. Information, robust influence, and residuals come from scores. The support mass omits $\bar{I}_i$ because discrimination directs the average but does not create an observation.

G.3 FROZEN TEXT CONSTRUCTION AND EMBEDDING

The fourteen English benchmark descriptions and their paraphrases were generated on 2026-08-29 with `claude-fable-5`. The recorded generation instruction requested a single plain-text English paragraph of 2–4 sentences and roughly 40–80 words identifying the measured domain, ability types, item format, and answer language. A companion paraphrase instruction requested the same meaning in clearly different wording and sentence structure at a similar length, again as one plain-text English paragraph. All fourteen descriptions and their paraphrases were fixed before analysis and are reproduced in Section G.4.

The task demonstrations begin with five user-facing queries. Each was canonicalized once into the description register, with comparable length and facet structure. Section G.6 reproduces both the raw queries and the exact canonical forms used in the experiments. All ten strings were fixed before fitting and evaluation.

All text is embedded with the `Qwen/Qwen3-Embedding-8B` checkpoint through `SentenceTransformers`. The encoder runs in float16 on an NVIDIA L40S with maximum sequence length 1024 and batch size 8, then stores normalized float32 embeddings. Benchmark descriptions and task queries are encoded symmetrically as documents without an instruction prefix. The reported geometry uses these normalized vectors directly, with no centering or component removal.

G.4 BENCHMARK DESCRIPTIONS AND PARAPHRASES

The following texts are reproduced exactly as experimental inputs; their wording defines the semantic representation evaluated in this study.

AIME. *Description.* This benchmark measures olympiad-track competition mathematics using the 2024 American Invitational Mathematics Examination: thirty problems spanning algebra, combinatorics, number theory, and geometry, each demanding a multi-step derivation that ends in an integer answer between 0 and 999. It rewards sustained exact symbolic reasoning with no partial credit and no multiple-choice scaffolding.

Paraphrase. Built from the AIME 2024 contest, this evaluation poses thirty high-school olympiad-pipeline math problems across algebra, counting, number theory, and geometry. Every item requires a long chain of precise reasoning that must terminate in a single three-digit integer, so scores reflect exact competition mathematics ability rather than recognition among options.

AIME’25. *Description.* This benchmark measures olympiad-track competition mathematics using the 2025 American Invitational Mathematics Examination: thirty problems spanning algebra, combinatorics, number theory, and geometry, each demanding a multi-step derivation that ends in an integer answer between 0 and 999. It is the newer edition of the same contest series, reducing training-data contamination for recent models.

Paraphrase. Drawn from the AIME 2025 contest, this evaluation contains thirty olympiad-pipeline problems in algebra, counting, number theory, and geometry whose solutions are single integers from 0 to 999. As the most recent edition of the contest series it probes the same exact competition-math skill while limiting exposure through training data.

MATH-500. *Description.* This benchmark measures high-school competition mathematics on a curated 500-problem subset of the MATH dataset, covering algebra, geometry, precalculus, probability, and number theory at difficulty levels from routine to contest-hard. Problems are free-response with exact symbolic or numeric answers, testing stepwise mathematical problem solving below olympiad intensity.

Paraphrase. A 500-item selection from the MATH corpus, this evaluation spans school-competition topics — algebra, geometry, precalculus, probability, number theory — across graded difficulty tiers. Each problem requires producing an exact final answer in free form, measuring general mathematical problem-solving that sits beneath olympiad-level contests.

GPQA Diamond. *Description.* This benchmark measures graduate-level scientific knowledge and reasoning with GPQA Diamond: 198 multiple-choice questions in biology, physics, and chemistry authored by PhD-holding experts and constructed to be Google-proof, so that skilled non-experts with web access still fail. Success requires deep domain understanding rather than retrieval or surface pattern matching.

Paraphrase. GPQA Diamond consists of 198 expert-written multiple-choice questions from biology, physics, and chemistry, deliberately designed so that even skilled searchers without domain training answer poorly. This evaluation therefore isolates genuine graduate-level scientific comprehension and inference rather than lookup ability.

HLE. *Description.* This benchmark measures frontier academic knowledge and reasoning with Humanity’s Last Exam: roughly 2,500 extremely difficult questions written by subject experts across mathematics, natural sciences, engineering, medicine, humanities, and more, in multiple-choice and exact-answer formats. It targets the hardest expert-level material, where even leading systems answer mostly incorrectly.

Paraphrase. Humanity’s Last Exam gathers around 2,500 exceptionally hard expert-authored questions covering mathematics, the sciences, engineering, medicine, and humanities, posed as multiple-choice or exact-answer items. It probes the outer edge of academic expertise, a regime where most current systems fail on the majority of items.

MMLU-Pro. *Description.* This benchmark measures broad academic and professional knowledge with MMLU-Pro: about twelve thousand ten-option multiple-choice questions across fourteen domains including law, medicine, engineering, mathematics, and business. The enlarged option set and reasoning-focused question selection make it a strengthened successor to MMLU, testing knowledge application over recognition.

Paraphrase. MMLU-Pro poses roughly 12,000 multiple-choice questions with ten answer options each, spanning fourteen academic and professional fields such as law, medicine, engineering, and business. Compared with its predecessor it emphasizes reasoning-heavy items and harder distractors, so it gauges applied breadth of knowledge rather than option recognition.

LiveCodeBench. *Description.* This benchmark measures competitive programming ability with LiveCodeBench: algorithmic problems continuously collected from LeetCode, AtCoder, and Codeforces after model training cutoffs to control contamination. Systems must generate correct, efficient Python solutions judged by hidden test cases, testing algorithm design, data structures, and precise implementation.

Paraphrase. LiveCodeBench sources fresh algorithm-contest problems from platforms like LeetCode, AtCoder, and Codeforces, dated after training cutoffs so memorization cannot help. Solving them demands designing algorithms and data structures and writing exact Python implementations that pass hidden tests, measuring genuine competitive-programming skill.

SciCode. *Description.* This benchmark measures scientific computing ability with SciCode: research-grade programming problems authored by scientists, requiring implementation of numerical methods and domain computations from physics, chemistry, biology, and materials science in Python. Problems decompose into subtasks that combine domain knowledge with correct numerical code.

Paraphrase. SciCode presents scientist-written programming challenges where the task is to code numerical routines and domain-specific computations — drawn from physics, chemistry, biology, and materials research — in Python. Each main problem breaks into dependent subproblems, jointly testing scientific understanding and the ability to translate it into working numerical code.

IFBench. *Description.* This benchmark measures precise instruction following with IFBench: prompts carrying explicit, programmatically verifiable constraints on the output, such as required formats, lengths, keyword inclusion or avoidance, and structural rules. Scoring checks constraint satisfaction exactly, so it isolates a system’s ability to obey specifications rather than produce generally good text.

Paraphrase. IFBench evaluates whether a system can follow exact output specifications: each prompt imposes verifiable requirements on format, length, wording, or structure, and the response is checked mechanically against them. The measure thus captures disciplined specification compliance, independent of overall answer quality.

Long-context reasoning. *Description.* This benchmark measures long-context reasoning with Artificial Analysis LCR: questions that require locating, connecting, and reasoning over information dispersed across very long multi-document inputs on the order of a hundred thousand tokens. It tests retrieval within context, cross-document synthesis, and reasoning that survives extreme prompt lengths.

Paraphrase. The AA long-context reasoning suite feeds systems inputs of roughly hundred-thousand-token scale built from multiple documents, then asks questions whose answers demand finding scattered evidence

and combining it correctly. Performance reflects in-context retrieval, synthesis across documents, and reasoning robustness at extreme context lengths.

Terminal-Bench Hard. *Description.* This benchmark measures autonomous command-line operation with the hard subset of Terminal-Bench: agentic tasks where a system controls a Unix shell over many steps to accomplish goals such as building software, manipulating files, configuring tools, and debugging environments. Success requires planning, tool use, and recovery from errors without human help.

Paraphrase. In Terminal-Bench’s hard split, a system is dropped into a Unix terminal and must autonomously complete multi-step objectives — compiling projects, wrangling files, configuring utilities, diagnosing failures — through repeated shell commands. The score captures agentic planning, tool operation, and self-correction in a real command-line environment.

Terminal-Bench 2.1. *Description.* This benchmark measures autonomous command-line operation with Terminal-Bench version 2.1: a revised edition of the agentic terminal suite where a system controls a Unix shell over many steps to build software, manipulate files, configure tools, and debug environments. The newer revision refreshes and rebalances the task set of the same underlying evaluation.

Paraphrase. Terminal-Bench 2.1 is the updated release of the agentic terminal evaluation: the system operates a Unix shell autonomously across many steps to complete goals like building projects, editing files, setting up tools, and fixing broken environments. It renews the task collection of the same benchmark family while measuring the same shell-agent capability.

τ^2 -Bench. *Description.* This benchmark measures conversational tool-using agents with tau-squared-bench: a system acts as a customer-service agent across domains such as airline, retail, and telecom, conversing with a simulated user while calling APIs and obeying domain policies. Scoring checks whether the final database state and responses satisfy the task, testing dialogue, tool orchestration, and policy compliance.

Paraphrase. In tau-squared-bench the system plays a customer-support agent for domains like airlines, retail, and telecom: it must talk with a simulated customer, invoke the right backend API calls, and respect business policies, with success judged by the resulting system state. The evaluation captures multi-turn dialogue management combined with correct, rule-abiding tool use.

τ -Bench Banking. *Description.* This benchmark measures conversational tool-using agents in the banking domain of the tau-bench family: the system serves as a banking customer-service agent, handling requests like transfers, card issues, and account changes by conversing with a simulated user, calling banking APIs, and following strict financial policies. It probes the same agentic skill set within a single regulated domain.

Paraphrase. The banking split of the tau-bench agent evaluation casts the system as a bank’s support agent: through dialogue with a simulated customer it must execute operations such as transfers and card management via API calls while strictly observing financial rules. It exercises conversational tool orchestration and policy compliance in one regulated vertical.

G.5 QUERY CANONICALIZATION

Table 11: Effect of query canonicalization. The support coefficient is the median over models of $c_m(q) = P_m(q)/(P_m(q) + 1)$.

Task query	Distance to nearest benchmark ↓		Median support factor	
	One sentence	Canonical	One sentence	Canonical
customer support automation	1.01	0.81	0.18	0.31
long report analysis	0.92	0.75	0.25	0.34
math competition coach	0.70	0.59	0.22	0.32
poetry writing partner	1.06	1.00	0.12	0.21
software engineering agent	0.99	0.70	0.20	0.39
Median benchmark–benchmark distance	0.89			

Benchmark descriptions share a template that names the measured subject, format, and difficulty. Canonicalization rewrites a user’s query into that register before embedding. The four queries with a corresponding benchmark family move closer to the list, while the poetry query remains the farthest. The resulting support coefficient follows that geometric proximity and sets the regularization strength of the query adjustment.

G.6 TASK-QUERY TEXTS

Software Engineering Agent. *Query.* Act as an autonomous software engineering agent that works in a terminal: plan multi-step changes, run build and test commands, edit files, and debug failures in a real repository without human intervention.

Canonical form. This task measures autonomous software engineering in a real repository: the system operates a terminal over many steps, planning changes, editing files, running build and test commands, and debugging failures without human intervention. Success requires agentic planning, tool use, code comprehension, and recovery from errors across long multi-step sessions.

Math Competition Coach. *Query.* Coach students preparing for mathematics olympiad qualifiers, solving competition problems in algebra, combinatorics, number theory, and geometry with rigorous step-by-step derivations.

Canonical form. This task measures coaching students toward mathematics olympiad qualifiers: solving competition problems in algebra, combinatorics, number theory, and geometry with rigorous step-by-step derivations, and explaining solution strategies clearly. It demands exact multi-step symbolic reasoning at contest level together with clear mathematical exposition.

Customer Support Automation. *Query.* Automate customer support for a subscription business: converse with customers, look up accounts, execute refunds and plan changes through internal APIs, and strictly follow company policy.

Canonical form. This task measures automated customer support for a subscription business: the system converses with customers in multi-turn dialogue, looks up accounts, executes refunds and plan changes through internal APIs, and strictly follows company policy. It requires conversational tool orchestration, policy compliance, and correct completion of business operations judged by the resulting account state.

Long Report Analysis. *Query.* Analyze bundles of lengthy regulatory filings and technical reports, answering questions that require finding and connecting evidence spread across hundreds of pages.

Canonical form. This task measures analysis of bundles of lengthy regulatory filings and technical reports: the system answers questions whose supporting evidence is dispersed across hundreds of pages spanning multiple documents. It requires retrieval within very long contexts, synthesis across documents, and faithful reasoning over the assembled evidence.

Poetry Writing Partner. *Query.* Collaborate with a poet on drafting and revising poems, offering imagery, meter, and line-break suggestions in an emotionally attuned voice.

Canonical form. This task measures collaborative poetry writing: drafting and revising poems with a human partner, offering suggestions on imagery, meter, and line breaks in an emotionally attuned voice. It requires literary creativity, aesthetic judgment, and sensitivity to tone, abilities not tied to factual knowledge or technical problem solving.

G.7 PER-BENCHMARK CURVE PARAMETERS

Table 12: Fitted characteristic-curve parameters and semantic weights for the fourteen Artificial Analysis benchmarks. The columns report discrimination a_i , difficulty b_i , residual scale σ_i , curve R^2 , and normalized semantic weight v_i .

Benchmark	a_i	b_i	σ_i	Curve R^2	v_i
AIME	1.81	-0.42	0.097	0.90	0.054
AIME'25	1.33	-0.29	0.126	0.83	0.052
GPQA Diamond	0.58	-0.80	0.064	0.90	0.078
HLE	0.79	+1.80	0.040	0.90	0.078
IFBench	0.44	+0.15	0.093	0.71	0.080
Long-context reasoning	0.86	+0.45	0.117	0.82	0.080
LiveCodeBench	0.92	-0.29	0.076	0.91	0.079
MATH-500	1.24	-1.56	0.062	0.92	0.068
MMLU-Pro	0.59	-1.48	0.057	0.88	0.078
SciCode	0.36	+1.36	0.067	0.77	0.079
τ^2 -Bench	1.10	+0.05	0.154	0.76	0.074
τ -Bench Banking	0.99	+1.88	0.055	0.84	0.074
Terminal-Bench Hard	0.86	+1.36	0.065	0.85	0.063
Terminal-Bench 2.1	1.33	+0.90	0.079	0.93	0.063
Equal weighting					0.071

AIME and AIME'25 receive semantic weights 0.054 and 0.052, jointly 0.106 where two equal-weight benchmarks would receive 0.143. This is the normalized aggregate counterpart of their reduced marginal mass in Figure 3.

G.8 FITTING DETAILS

Alternating fit. Calibration alternates for six rounds. Given model abilities, each benchmark's (a_i, b_i) is fitted by weighted least squares on its observed column. Given the curves, each θ_m is fitted over $[-3.5, 3.5]$ in steps of 0.02 using Equation (2), then refined by a parabola through the minimum grid point and its two neighbours. Abilities are standardized to mean zero and unit variance.

Robust cell weight. The weight $w_{mi} = \min(1, (3/|r_{mi}|)^2)$ is floored at 10^{-3} and applied from the second round. Residuals within three standard deviations retain unit weight. Larger residuals are attenuated quadratically. The same weights enter the ability fit, residual field, and support mass.

G.9 GLOBAL-RANKING COMPARISON

Across the 584 model configurations with both scores, the two full rankings agree at Spearman $\rho = 0.967$ and Kendall $\tau = 0.855$. The *Observed* column counts non-missing scores among the fourteen analyzed benchmarks.

Table 13: Union of the top 20 model configurations under the BoB global level score θ_m and the AA Intelligence Index. Bold indicates top-20 membership.

Model configuration	Benchmarks observed	BoB rank	θ_m	AA rank	AA Index
Claude Opus 5 (Adaptive Reasoning, Xhigh Effort)	6	1	1.734	2	62.5
Claude Opus 5 (Adaptive Reasoning, Max Effort)	6	2	1.731	1	63.1
Claude Opus 5 (Adaptive Reasoning, High Effort)	6	3	1.730	4	61.5
Kimi K3 (max)	6	4	1.725	8	59.7
GPT-5.6 Sol (max)	9	5	1.709	5	60.9
Grok 4.6 (high)	6	6	1.703	5	60.9
GLM-5.3 (max)	6	7	1.675	9	59.5
Claude Fable 5 (Adaptive Reasoning, Max Effort, Opus 4.8 Fallback)	9	8	1.668	3	62.1
Qwen3.8 Max	6	9	1.649	13	58.1
Claude Opus 5 (Adaptive Reasoning, Medium Effort)	6	10	1.640	12	58.6
Grok 4.6 (xhigh)	6	11	1.631	7	60.0
Qwen3.8 2.4T A95B	6	12	1.628	14	57.7
GPT-5.5 (xhigh)	9	13	1.625	20	56.3
GPT-5.6 Sol (xhigh)	9	14	1.618	10	59.0
GLM-5.3-Flash	6	15	1.595	15	57.5
Gemini 3.7 Flash (high)	6	16	1.594	21	56.0
Grok 4.6 (medium)	6	17	1.593	10	59.0
GPT-5.6 Terra (max)	9	18	1.588	19	56.6
GPT-5.6 Sol (high)	9	19	1.582	16	57.3
Qwen3.8-Flash-Next	6	20	1.576	22	55.8
Muse Spark 1.2 (xhigh)	6	21	1.563	18	56.8
Claude Opus 4.8 (Adaptive Reasoning, Max Effort)	9	26	1.528	16	57.3