Submodular Benchmark Selection

Alex Smola
Boson AI
alex@smola.org

Abstract

Evaluating a new language model means running it on a large, growing collection of benchmarks, at considerable cost. Much of this cost is redundant: arranged as a matrix of models by benchmarks, scores are strongly correlated and concentrate in a few dimensions, so a few well-chosen benchmarks largely determine the rest. This allows us to run a new model on a few benchmarks and predict the rest with a simple Gaussian model that copes with missing scores. The key question is *which few to run*. An answer comes from sensor placement, which chooses the observations that carry the most information, measured by entropy or mutual information. We argue that the objective should instead reward a selection for how well it predicts every benchmark. This leads to two coverage objectives: explained variance, the predictions' accuracy if the estimated correlations were exact, and facility location, which credits each benchmark with its best proxy among the selected ones. On eleven score matrices built from public evaluation results, the coverage objectives predict the scores of held-out models best wherever the correlations can be estimated. The information objectives fall behind because they favor benchmarks that the selection predicts poorly (entropy) or that the others predict well (mutual information), not benchmarks that predict the others. By contrast, facility location predicts practically as well as explained variance, holds up on the addition of newer models and under real costs, has classical guarantees (even under a budget), and gives readable selections.

`https://github.com/smolix/benchmark-selection`

1 Introduction

Language models are judged by their scores on many benchmarks. Leaderboards such as HELM [Liang et al., 2023a] and the Open LLM Leaderboard [Beeching et al., 2023, Open LLM Leaderboard Team, 2024] run every model on dozens of tasks, the suites keep growing, and the cost recurs for every new model, checkpoint and ablation.

Much of this cost is redundant. A poor-quality model is likely to underperform on every benchmark. Arrange the scores in a matrix with one row per model and one column per benchmark. The columns are strongly correlated (a model that does well in college chemistry tends to do well in high-school chemistry), and the matrix is nearly low-rank. Figure 1 shows this for three score matrices that we assembled from public results: the accuracies of 5,452 language models on the 57 subjects of the knowledge test MMLU [Hendrycks et al., 2021], which we collected from the first version of the Open LLM Leaderboard [Beeching et al., 2023]; the scores of 263 text-embedding models on 56 tasks of the Massive Text Embedding Benchmark (MTEB) [Muennighoff et al., 2023]; and the scores of 118 models on 114 benchmarks that we pooled from nine leaderboards by matching model names. For these matrices, 2, 6 and 8 principal components, respectively, explain 90% of the variance. A few well-chosen benchmarks should suffice to determine a model's scores.

This suggests a cheaper evaluation: run a new model on a few selected benchmarks and predict its other scores with a deliberately simple model, the conditional mean of a multivariate Gaussian, which

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

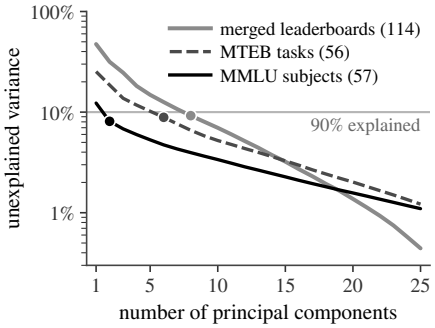


Figure 1: Benchmark scores are nearly low-rank, so a few benchmarks should reveal the rest. Share of variance that the leading principal components of the benchmarks' correlation matrix leave unexplained, against their number.

copes with missing scores and needs only the correlations between benchmarks, estimated on the models evaluated before (Section 2).

A classical answer to benchmark selection comes from sensor placement, which puts a few sensors where their readings carry the most information about a spatial field [Guestrin et al., 2005, Krause et al., 2008b], measured by the entropy of the readings or their mutual information (MI) with the readings elsewhere. Yet, information is not prediction error. For prediction, every benchmark needs a good *proxy* among those run, a selected benchmark that predicts it well, and running one of two near-duplicate chemistry subjects predicts both. Entropy instead favors a benchmark that the selected ones predict poorly, such as one that resembles no other and so predicts no other; MI favors a benchmark that the others predict well, such as a second near-duplicate or an average of several benchmarks, whether or not it stands in for them.

We therefore derive the selection objective from the metric by which predictions are judged: R^2, the share of the variance of the unselected scores that the predictions explain. If the estimated correlations were exact, a selection's R^2 would be fixed by the variance that the selected benchmarks explain, summed over all benchmarks. We call this sum *explained variance* (EV); it extends the classical criterion of subset selection for regression [Das and Kempe, 2011] to many targets, and maximizing it maximizes that R^2. Crediting each benchmark only with its best proxy among the selected ones gives *facility location* (FL), which Yao et al. [2026] used to select benchmarks; with squared correlations as similarities, it is a lower bound on explained variance. Both are *coverage* objectives: they reward a selection for how well it predicts every benchmark. In short: coverage, not information.

On 11 score matrices built from public evaluation results, compared on shared splits of the models into training and held-out models, the coverage objectives predict best. Facility location is never significantly worse than entropy, MI or three simple baselines and is significantly better than each in about half to two-thirds of the comparisons; explained variance is significantly ahead of it in a few, by at most 0.005 in R^2. Facility location stays at or near the top on models released later and under real costs. Selection pays most when only a few benchmarks fit the budget. We recommend facility location: it predicts practically as well as explained variance; greedy selection, which adds one benchmark at a time, provably reaches at least $1 - 1/e$ of its optimum (a cost-aware variant half of that under a budget); it uses only pairwise correlations, except to break exact ties; and each selected benchmark is the best proxy for the benchmarks it covers, so selections are easy to check.

Our contributions are (i) a derivation of selection objectives from the prediction metric, which shows why the information objectives do not follow from it (Sections 2 and 3); (ii) a paired comparison of seven selectors on 11 score matrices, including models released later and per-benchmark costs (Sections 4, 5 and 7); (iii) an explanation of the negative result for the information objectives (Section 6); and (iv) a practical recommendation, with code and data.

2 Predicting scores from a few benchmarks

We first fix how the unselected scores are predicted and judged, because the selection objectives of Section 3 follow from both.

Setup, predictor and metric. The score matrix $B \in \mathbb{R}^{M \times N}$ holds the scores B_{mj} of M models on N benchmarks. Many scores are typically missing, since leaderboards rarely run every model on every benchmark. We split the models into *training models*, which stand for the models evaluated so

far, and *validation models*, held out to stand for models not yet evaluated. On the training models we standardize each benchmark, $z_{mj} = (B_{mj} - \hat{\mu}_j)/\hat{\sigma}_j$, and estimate the correlation matrix $\hat{R}$ of the benchmarks. A new model runs only a selected set $\mathcal{A} \subseteq [N] = \{1, \ldots, N\}$ of k benchmarks, and its scores on the unselected benchmarks $\bar{\mathcal{A}} = [N] \setminus \mathcal{A}$ are to be predicted. We treat a model's standardized scores as a draw from a Gaussian distribution with mean zero and covariance $\hat{R}$, and predict the unselected scores by their conditional mean given the selected ones, $\hat{z}_{\bar{\mathcal{A}}} = \hat{R}_{\bar{\mathcal{A}}\mathcal{A}}\hat{R}_{\mathcal{A}\mathcal{A}}^{-1} z_{\mathcal{A}}$: a linear regression on the selected benchmarks, with coefficients taken from $\hat{R}$. For a validation model m, let $\mathcal{A}_m \subseteq \mathcal{A}$ be the selected benchmarks it has scores on and $T_m \subseteq \bar{\mathcal{A}}$ its observed unselected benchmarks, the *targets*. We predict the targets with a small ridge $\lambda = 10^{-2}$, clip each prediction to its benchmark's valid range, and pool the squared errors over all validation models:

$$\hat{z}_{m,T_m} = \hat{R}_{T_m\mathcal{A}_m}\left(\hat{R}_{\mathcal{A}_m\mathcal{A}_m} + \lambda I\right)^{-1} z_{m,\mathcal{A}_m}, \qquad R^2(\mathcal{A}) = 1 - \frac{\sum_m \sum_{j\in T_m} (z_{mj} - \hat{z}_{mj})^2}{\sum_m \sum_{j\in T_m} z_{mj}^2}. \quad (1)$$

An R^2 of one means perfect predictions; zero means no better than the training mean ($z = 0$).

Why a linear Gaussian model. We chose this simple model for four reasons. First, it needs only means and correlations, which can be estimated from incomplete score tables, pairwise (each correlation from the models that report both scores) or by expectation maximization (EM) [Dempster et al., 1977, Little and Rubin, 2019], and it predicts from whichever selected scores a model has. Second, the same formula with the true correlations R and no ridge, $R_{\bar{\mathcal{A}}\mathcal{A}}R_{\mathcal{A}\mathcal{A}}^{-1} z_{\mathcal{A}}$, is the linear predictor with the smallest expected squared error for every distribution with mean zero and covariance R, Gaussian or not. Squared error is also how we judge the predictions, so we need no normality, which tests reject for most benchmarks (Appendix B.6). Third, the predictor, the R^2 it can be expected to reach (Section 3) and every selection objective are functions of the same correlation matrix, so the whole pipeline is coherent and in closed form. Fourth, the predictor is simple and the same for every selector, so differences in R^2 reflect the selections, not a flexible predictor that makes up for a poor choice. Flexible predictors also did not help: low-rank matrix completion [Mazumder et al., 2010] was not significantly better on facility location's selections, the pretrained imputer TabImpute [Feitelberg et al., 2025] was far worse, and logit-transformed scores did not help the coverage objectives (Appendices G.5 and G.7).

3 From the metric to selection objectives

We want to find the k benchmarks that maximize the R^2 on validation models. R^2 is unknown when the benchmarks are chosen, since the validation scores are held out, but we can compute the R^2 that a selection would reach if $\hat{R}$ were the true correlation matrix, its *plug-in* R^2, and turn it into objectives.

Greedy selection. Searching all sets of k benchmarks is infeasible beyond small k, so we select greedily: starting from the empty set, we add one benchmark at a time, each time the candidate j with the largest gain $\Delta(j|\mathcal{A}) = F(\mathcal{A} \cup \{j\}) - F(\mathcal{A})$ in an objective F, until k are selected. Greedy selection is provably near-optimal when F is normalized ($F(\emptyset) = 0$), *monotone*, so that a pick never lowers it, and *submodular*, so that gains diminish as the selection grows: $\Delta(j|\mathcal{A}) \geq \Delta(j|\mathcal{A}')$ for $\mathcal{A} \subseteq \mathcal{A}' \not\ni j$. The greedy set then reaches at least $1 - 1/e \approx 0.63$ of the best value over all sets of size k [Nemhauser et al., 1978], a factor that no efficient algorithm can improve in general [Feige, 1998]. Submodularity formalizes redundancy: once one of two near-duplicate chemistry subjects is selected, the other adds little.

Explained variance. If a model's standardized scores have covariance R (a correlation matrix; $\hat{R}$ in practice) and it has run the benchmarks in $\mathcal{A}$, the errors of the predictor of Section 2 (without ridge) on the unselected benchmarks have the *residual covariance* $S(\mathcal{A}) = R_{\bar{\mathcal{A}}\bar{\mathcal{A}}} - R_{\bar{\mathcal{A}}\mathcal{A}}R_{\mathcal{A}\mathcal{A}}^{-1}R_{\mathcal{A}\bar{\mathcal{A}}}$. For each benchmark j, $\sigma^2_{j|\mathcal{A}} = 1 - R_{j\mathcal{A}}R_{\mathcal{A}\mathcal{A}}^{-1}R_{\mathcal{A}j}$ is the variance that a linear regression on the benchmarks in $\mathcal{A}$ leaves unexplained (a diagonal entry of S for $j \notin \mathcal{A}$), and $R^2(j|\mathcal{A}) = 1 - \sigma^2_{j|\mathcal{A}}$ is the variance it explains (1 for $j \in \mathcal{A}$, 0 for $\mathcal{A} = \emptyset$, and the squared correlation R^2_{ij} for $\mathcal{A} = \{i\}$). A good selection makes S small, and the four objectives differ in how they measure its size. Note that squared error measures it by its trace. As such, *explained variance*, $F_{\mathrm{EV}}(\mathcal{A}) = \sum_{j=1}^{N} R^2(j|\mathcal{A}) = N - \operatorname{tr} S(\mathcal{A})$, is the metric in disguise.

Proposition 1 (Explained variance determines R^2) *Let R be a positive definite correlation matrix (unit diagonal), and suppose that a new model's standardized scores $z \in \mathbb{R}^N$ have mean 0 and covariance R, are fully observed, and are predicted by $\hat{z}_{\bar{\mathcal{A}}} = R_{\bar{\mathcal{A}}\mathcal{A}} R_{\mathcal{A}\mathcal{A}}^{-1} z_{\mathcal{A}}$ (no ridge, no clipping). Then, for $0 < k < N$,*

$$R^2_{\mathrm{pop}}(\mathcal{A}) := 1 - \frac{\mathbb{E}\|z_{\bar{\mathcal{A}}} - \hat{z}_{\bar{\mathcal{A}}}\|^2}{\mathbb{E}\|z_{\bar{\mathcal{A}}}\|^2} = \frac{1}{N-k} \sum_{j \in \bar{\mathcal{A}}} R^2(j|\mathcal{A}) = \frac{F_{\mathrm{EV}}(\mathcal{A}) - k}{N-k}. \tag{2}$$

The identity is well known: Das and Kempe [2011] maximize one of its terms, R^2 of a single target, and we include a short proof in Appendix A for completeness. Only means and covariances enter it. Evaluated at $\hat{R}$, R^2_{pop} is the plug-in R^2, so greedy explained variance maximizes the plug-in R^2 one benchmark at a time. It should do best where $\hat{R}$ is accurate; where it is not, the plug-in value can overstate the *realized* R^2 measured on validation models (Section 5).

A pairwise approximation: facility location. Explained variance has two drawbacks. It is not submodular, because a benchmark that explains little alone can explain more next to another (Appendix A), so greedy selection has only a data-dependent guarantee; and its gains need the inverse $\hat{R}^{-1}_{\mathcal{A}\mathcal{A}}$, which amplifies the error of correlations estimated from few or partly observed models. Both drawbacks disappear if each benchmark is credited only with what its best single proxy in $\mathcal{A}$ explains. With squared correlations as similarities, this is *facility location* [Cornuejols et al., 1977], named after the problem of opening a few facilities so that every customer is close to one; here the benchmarks are the customers and the selected benchmarks the facilities. It also turns out to be a (mathematically more amenable) lower bound on explained variance.

Proposition 2 (A pairwise lower bound on explained variance) *Facility location with squared similarities, $F^{\mathrm{sq}}_{\mathrm{FL}}(\mathcal{A}) := \sum_{j=1}^{N} \max_{i \in \mathcal{A}} R^2_{ij}$, satisfies $F^{\mathrm{sq}}_{\mathrm{FL}} \le F_{\mathrm{EV}}$, with equality if $|\mathcal{A}| \le 1$.*

A joint regression explains at least what its best single member explains; the difference is what proxies add in combination. Facility location is monotone and submodular, since a new proxy adds only where it beats the best one already selected, and its gains need only pairwise correlations. Yao et al. [2026] use it to select benchmarks by their score correlations, with the clipped similarity $r^+_{ij} = \max(R_{ij}, 0)$, and we adopt their objective $F_{\mathrm{FL}}(\mathcal{A}) = \sum_{j=1}^{N} c_j(\mathcal{A})$, where the *cover* $c_j(\mathcal{A}) = \max_{i \in \mathcal{A}} r^+_{ij}$ (zero for $\mathcal{A} = \emptyset$) is the clipped correlation of j with its best proxy. We keep their similarity, fixed before any experiment, to test their method as published; while it gives no bound on explained variance (unlike squared similarities), in practice it is essentially indistinguishable from squared similarities. See Appendix G.6 for a comparison of the two scores.

Information-theoretic objectives. Sensor placement [Guestrin et al., 2005, Krause et al., 2008b], where sensors play the role of benchmarks and a kernel gives the covariance that we estimate from past models, measures S by its determinant instead. For Gaussian scores and fixed k, maximizing the *entropy* of the selected scores, a measure of how uncertain they are jointly, amounts to maximizing $F_{\mathrm{H}}(\mathcal{A}) = \log\det R_{\mathcal{A}\mathcal{A}} = \log\det R - \log\det S$. The *mutual information* $F_{\mathrm{MI}}(\mathcal{A}) = \frac{1}{2}\log\big(\det R_{\mathcal{A}\mathcal{A}} \det R_{\bar{\mathcal{A}}\bar{\mathcal{A}}} / \det R\big) = \frac{1}{2}\log\det R_{\bar{\mathcal{A}}\bar{\mathcal{A}}} - \frac{1}{2}\log\det S$ measures how much the selected scores tell about the unselected ones. Both are submodular. Explained variance measures how much residual variance a selection removes; entropy and MI measure only the factor by which it shrinks, so halving a tiny residual variance counts as much as halving a large one, although only the large one matters for squared error.

To make this more concrete, consider three near-duplicate chemistry subjects, two benchmarks that correlate weakly with them, a slightly noisy average of these two, and an *outlier* that correlates with no other benchmark. The coverage objectives start with a chemistry subject, which stands in for its two twins, and then add the average, which stands in for both of its parts; entropy makes the same first pick (by the tie rule below) but then adds the outlier, which the selection predicts worst yet which stands in for no benchmark but itself. MI starts with the average, which the other benchmarks predict best jointly, and its third pick is a second chemistry subject, redundant with the first but well predicted by the third. In the first step MI ranks j by $R^2_{j|\mathrm{rest}} = R^2(j \mid [N] \setminus \{j\})$, how well the others jointly predict j, quite different from the coverage objectives (Table 1).

Table 1: Selection objectives: coverage of every benchmark (facility location, FL; explained variance, EV) or information in the selected scores (entropy; mutual information, MI). Only the coverage objectives score a benchmark by what it predicts. c_i: cover; $\sigma^2_{j|\mathcal{A}}$: unexplained variance; $R^2_{j|\mathrm{rest}}$: variance of j that all others explain; γ: submodularity ratio (Proposition 3).

	First-step gain of j	Greedy then adds the benchmark that …	Greedy guarantee
FL	$1+\sum_{i\neq j} r^+_{ij}$	most raises the covers $c_i(\mathcal{A})$	$1-1/e$, knapsack $\frac{1}{2}(1-1/e)$
EV	$1+\sum_{i\neq j} R^2_{ij}$	explains most remaining variance jointly	$1-e^{-\gamma}$
Entropy	0 for every j (tie)	$\mathcal{A}$ predicts worst (largest $\sigma^2_{j\|\mathcal{A}}$)	only after an arbitrary shift
MI	$-\frac{1}{2}\log\bigl(1-R^2_{j\|\mathrm{rest}}\bigr)$	$\mathcal{A}$ predicts poorly, the rest predict well	e^{-1} for random greedy

Proposition 3 (Greedy guarantees) *Let $\mathcal{A}_G$ be the greedy set of size k, and OPT_k the best value of the same objective over sets of at most k benchmarks.*

1. *As $r^+ \geq 0$, F_{FL} is normalized, monotone and submodular, so $F_{\mathrm{FL}}(\mathcal{A}_G) \geq (1-1/e)\,\mathrm{OPT}_k$ [Nemhauser et al., 1978].*
2. *Under costs $\kappa_j > 0$ and a budget C,* knapsack selection *returns the better of cost-effective greedy (add the affordable benchmark with the largest gain per unit cost) and the best affordable single benchmark; for F_{FL} it reaches $\frac{1}{2}(1-1/e)$ of the best affordable set [Krause and Guestrin, 2005].*
3. *F_{EV} is normalized and monotone but not submodular; $F_{\mathrm{EV}}(\mathcal{A}_G) \geq (1-e^{-\gamma})\,\mathrm{OPT}_k$ with a submodularity ratio $\gamma \in [0,1]$, which measures how far F_{EV} is from submodular ($\gamma = 1$ if it is) and depends on R and k [Das and Kempe, 2011, Bian et al., 2017].*
4. *$F_{\mathrm{MI}} \geq 0$ is submodular and symmetric, $F_{\mathrm{MI}}(\mathcal{A}) = F_{\mathrm{MI}}(\bar{\mathcal{A}})$, hence not monotone unless it vanishes: deterministic greedy has no known constant-factor guarantee, only a bound up to an additive loss [Krause et al., 2008b], while Random Greedy, a randomized variant, achieves $\mathbb{E}\,F_{\mathrm{MI}}(\mathcal{A}) \geq \mathrm{OPT}_k/e$ [Buchbinder et al., 2014].*
5. *F_{H} is submodular, but its gains $\log \sigma^2_{j|\mathcal{A}}$ are at most 0, so a bound needs an arbitrary shift $F_{\mathrm{H}}(\mathcal{A}) + \delta|\mathcal{A}|$ and weakens as δ grows.*

Of the deterministic methods, only facility location thus has a guarantee that does not depend on the data, with or without a budget; Appendix A checks that these known results apply here and that greedy comes much closer to the optimum on our data. Entropy's first step is an exact tie ($\sigma^2_{j|\emptyset} = 1$ for every j); all selectors break near-ties by the explained-variance gain (per unit cost under a budget), then by index (Appendix B.3), which keeps the guarantees up to a negligible term. Note that all guarantees concern the objective on the estimate $\hat{R}$, not the realized R^2.

Baselines. Three simpler heuristics serve as baselines. *Hub* takes the k benchmarks with the largest mean absolute correlation $h_j = (N-1)^{-1}\sum_{i\neq j}|\hat{R}_{ij}|$, ranked once rather than greedily, so it ignores redundancy and would take both near-duplicates; its first pick is usually that of facility location. *PCA* takes the k benchmarks that best span the top k principal components of $\hat{R}$, by column-pivoted QR [Businger and Golub, 1965], and *random* takes k benchmarks uniformly at random.

4 Experimental protocol

Our experiments answer three questions: which objective best predicts the scores of validation models (Section 5), why the information objectives fall behind (Section 6), whether the selected benchmarks still work for models released after the selection, and whether selection works when benchmarks differ in cost (Section 7). Good selectors often differ by less than the split-to-split variation of one selector's R^2, so every comparison is paired.

Score matrices. We call the three matrices of Figure 1 MMLU (fully observed), MTEB (77.3% observed) and MERGED (31.2% observed). With BenchPress [Papailiopoulos, 2026, Zeng and Papailiopoulos, 2026], which compiles the scores of 83 models on 49 benchmarks from papers and model cards (33.8% observed), they are our four main matrices, from dense to sparse. We call

BenchPress *estimation-limited*: too few models share scores to estimate its correlations well. Seven extended matrices test generality: the sub-tasks of the Open LLM Leaderboard v2 (OLL v2) [Open LLM Leaderboard Team, 2024], the scores of 395 models from its first version on 62 benchmarks, with public per-question results (OLL v1 items) [Maia Polo et al., 2024], Epoch AI's benchmark data (Epoch) [Epoch AI, 2026], three HELM suites (HELM Lite, HELM MMLU, HELM Classic) [Liang et al., 2023a], and a 10/2026 version of MTEB that adds the models submitted after our 3/2026 snapshot (MTEB-2026) [MTEB contributors, 2026]. Appendix C provides details.

Matched splits. Five random partitions of the models into ten folds give 50 splits per matrix; in each, one fold supplies the validation models and the other nine the training models. All methods share each split's training models, correlation estimate $\hat{R}$, candidate benchmarks and random numbers, so their R^2 values pair split by split. $\hat{R}$ is the pairwise correlation matrix if the training matrix is at least 90% observed and has at least as many models as benchmarks, and an EM estimate otherwise. EM assumes that scores are missing at random [Rubin, 1976] (Appendix B.2). A benchmark can be selected if at least three training models observe it (it is *eligible*) and serves as a target only if at least two do; we report $k \leq \min(15, N-3)$. On sparse matrices a validation model is predicted from the selected scores it has and skipped if it has none, so (1) measures prediction from the scores a leaderboard already holds, not deployment, where a new model would run every selected benchmark.

Paired statistics. We compare methods a and b through the per-split differences $d_s = R^2_{a,s} - R^2_{b,s}$ on the $n \leq 50$ splits where both are defined. Overlapping training sets correlate the d_s, so we use the 95% interval of Nadeau and Bengio [2003] for repeated cross-validation [Bouckaert and Frank, 2004], $\bar{d} \pm t_{0.975,\,n-1}\,\hat{\sigma}_d\sqrt{1/n + n_{\mathrm{val}}/n_{\mathrm{tr}}}$ with $\hat{\sigma}_d$ the standard deviation of the d_s, which at $n = 50$ and $n_{\mathrm{val}}/n_{\mathrm{tr}} \approx 1/9$ is 2.6 times as wide as the naive interval. A difference is *significant* if its interval excludes 0; we test each matrix and k separately, without correction for multiple comparisons.

5 Coverage objectives predict best

The two coverage objectives predict the unselected benchmarks best, and they are practically equal (Table 2 and Figure 2). Over all 11 matrices (bottom of Table 2), facility location is never significantly worse than MI, entropy, hub, PCA or random, and it is significantly better than each of them in 15 to 22 of 32 settings of matrix and k ($k \in \{3, 5, 10\}$, $k \leq N - 3$). We leave out $k = 1$, where several selectors make the same pick: facility location usually that of hub, and entropy by construction that of explained variance. The matrices overlap (MTEB-2026 extends MTEB, and three share the 57 MMLU subjects), so these are not independent confirmations. Facility location is never significantly better than explained variance and is significantly worse in 6 settings, all on the dense MMLU, OLL v2 and OLL v1 items matrices, by at most 0.005. The seven extended matrices follow the pattern of the main ones, and over every k the picture is the same (Table 4 in Appendix D). Selection pays most for small k: on MTEB, facility location leads random by 0.157 at $k = 1$ but by only 0.023, not significantly, at $k = 5$.

On the dense MMLU, selections can be read by hand. Every selector except hub predicts the unselected subjects well, and explained variance leads facility location by a consistent but negligible margin (at $k = 5$ in 50 of 50 splits, by 0.001). Facility location picks the same five subjects on all 50 splits: professional psychology and miscellaneous, the best proxies of 27 and 21 of the other subjects; elementary mathematics, the best proxy of 4; and college mathematics and abstract algebra, each the best proxy of no other subject, picked because the earlier picks cover them worst. Explained variance picks 4 of these subjects as well. Random selection is close behind because the scores are nearly low-rank (Figure 1): most random sets of five capture the few dominant directions. Hub falls behind as k grows because it keeps adding subjects that its earlier picks already cover.

On sparser matrices, the two coverage objectives stay tied. On MTEB explained variance has the higher mean for every $k \geq 2$, and on MERGED facility location does, but neither lead is significant at any k; facility location ties MI on MTEB and is significantly ahead of it on MERGED at $k = 5$ and 10 (Table 2). Where $\hat{R}$ is noisy, the two pick different benchmarks of similar quality: on MERGED they share only 21% of their picks at $k = 5$, against 80% on MMLU.

Table 2: Facility location (FL) is never significantly worse than the information objectives or the baselines, and practically equal to explained variance (EV). Top: mean R^2 over the 50 matched splits on the main matrices (others in Appendix D); bottom: settings over all 11 matrices ($k \in \{3, 5, 10\}$, $k \leq N - 3$) in which FL is significantly better, not different, or worse. $^{\triangledown}$/$^{\blacktriangle}$: significantly below/above FL (paired 95% interval, uncorrected); bold: best in the row; †: mean over the splits where a validation model observes the pick (at least 41 of 50).

Matrix	k	FL	EV	MI	Entropy	Hub	PCA	Random
MMLU	1	**0.860**	**0.860**	0.832$^{\triangledown}$	**0.860**	**0.860**	**0.860**	0.751$^{\triangledown}$
	5	0.919	**0.920**$^{\blacktriangle}$	0.906$^{\triangledown}$	0.913$^{\triangledown}$	0.875$^{\triangledown}$	0.907$^{\triangledown}$	0.890$^{\triangledown}$
	10	0.945	**0.946**$^{\blacktriangle}$	0.911$^{\triangledown}$	0.936$^{\triangledown}$	0.886$^{\triangledown}$	0.940$^{\triangledown}$	0.911$^{\triangledown}$
MTEB	1	**0.659**	0.646	0.650	0.646	**0.659**	0.648	0.502$^{\triangledown}$
	5	0.767	**0.790**	0.767	0.746	0.687$^{\triangledown}$	0.755	0.744
	10	0.826	**0.839**	0.821	0.817	0.718$^{\triangledown}$	0.801	0.784
MERGED	1	0.317	**0.490**	0.084†	**0.490**	0.320	0.489	0.260†
	5	**0.614**	0.593	0.465$^{\triangledown}$	0.503$^{\triangledown}$	0.481	0.483$^{\triangledown}$	0.563
	10	**0.661**	0.626	0.528$^{\triangledown}$	0.549$^{\triangledown}$	0.537	0.539$^{\triangledown}$	0.604
BenchPress	1	−0.033	−0.143	−0.021†	−0.143	−0.071	−0.136	**0.028**†
	5	0.053	−0.085	−0.059†	−0.300	0.032	0.060†	**0.120**
	10	0.066	−0.009	−0.088	−0.065	0.035	−0.049	**0.207**
Facility location against each column on 11 matrices, $k \in \{3, 5, 10\}$ (32 settings):								
FL significantly better		—	0	15	15	22	17	16
no significant difference		—	26	17	17	10	15	16
FL significantly worse		—	6	0	0	0	0	0

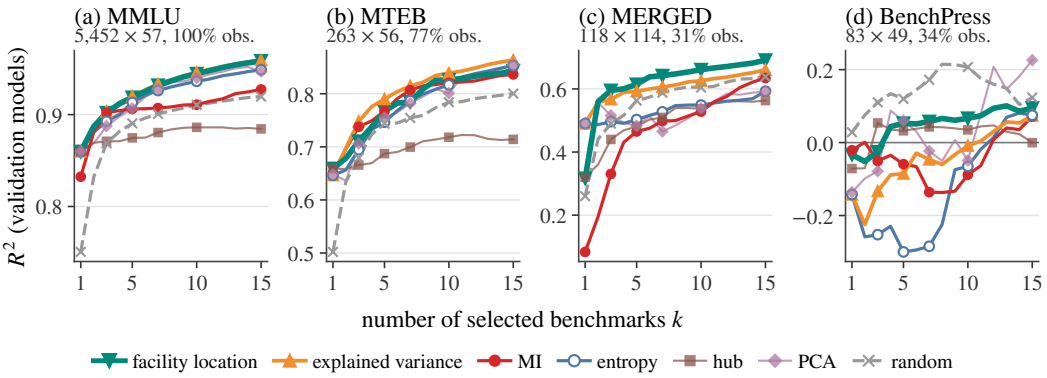


Figure 2: The coverage objectives predict best wherever the correlations can be estimated (a–c); on the estimation-limited BenchPress (d), no method significantly beats random. Mean R^2 on the validation models over the 50 matched splits (fewer where no validation model observes a pick; † in Table 2); significance in Table 2, intervals at $k = 5$ in Table 4.

Where correlations cannot be estimated, no selector helps. On BenchPress no selector is significantly better than random at any k; random even has the highest mean at $k = 5$ and 10, though not significantly. The limit is the correlation estimate, not the selector: at $k = 5$ the plug-in R^2 of explained variance (Proposition 1) is 0.756 here but its realized R^2 only −0.085, while on MMLU the two agree (0.921 and 0.920), so $\hat{R}$ cannot tell good proxies from poor ones (Appendix H). With a tenth of the MERGED models for training instead of nine tenths, every selector we ran falls to the level of random there too (Appendix D).

The information objectives fall behind. MI and entropy are never significantly better than facility location and are significantly worse in about half of the settings, also on the fully observed MMLU. An earlier version recommended MI because it beat entropy at small k; that comparison rested on entropy's first pick, an exact tie broken by rounding noise, and does not survive (Appendix I).

6 Why coverage beats information

The coverage objectives win because they reward what the metric rewards: good predictions of the left-out benchmarks; the information objectives behave as in the toy example of Section 3 (Appendix H). To see how well each objective guides selection, we draw random sets of five benchmarks and correlate each objective's value with the R^2 that the set actually reaches (Figure 3c). On MMLU both coverage objectives rank the sets as the metric does; on MERGED, where every objective is a weak guide, they track the realized R^2, while the information objectives do not.

Entropy rewards novelty, not coverage. After its tie-broken first pick, entropy adds the benchmark with the largest conditional variance $\sigma^2_{j|\mathcal{A}}$, the one that the selected benchmarks predict worst. Such a benchmark is unlike the selected ones, but nothing makes it a good proxy for the rest. Its sets are diverse (mean $|\hat{R}_{ij}|$ within its five picks 0.66 on MMLU, against 0.71 for facility location), but diversity within the selected set is not coverage of the rest.

MI rewards the predicted, not the predictor. MI's first pick maximizes $-\frac{1}{2}\log(1 - R^2_{j|\mathrm{rest}})$: it is the benchmark that all the others predict best jointly. Like the average in the toy suite, such a benchmark can be predicted well by the others whether or not it stands in for them. On MERGED, MI's usual first pick (48 of 50 splits) is the overall score of Arena-Hard-Auto [Li et al., 2024], a chat benchmark judged by a language model. It has $R^2_{j|\mathrm{rest}} = 0.95$ under $\hat{R}$, yet on its own it predicts the other benchmarks with $R^2 = 0.170$, against 0.533 for MMLU's astronomy subject, the usual first pick of explained variance (Figure 3b). The score is also fragile: it equals $\frac{1}{2}\log P_{jj}$ for the precision matrix $P = \hat{R}^{-1}$, and $P_{jj} = \sum_i v^2_{ij}/\lambda_i$, summed over the eigenvalues λ_i and eigenvectors v_i of $\hat{R}$, is dominated by the smallest eigenvalues, which few, partly observed models estimate worst. On the sparser matrices, MI's first pick draws nearly all of P_{jj} from the smallest tenth of the eigenvalues, and that of explained variance much less (Appendix H).

Neither the optimizer nor estimation noise alone explains the gap. If greedy were merely a poor optimizer of MI, Random Greedy, which unlike deterministic greedy has a constant-factor guarantee for MI (Proposition 3), should help; yet it reaches lower MI values and predicts no better (Appendix A). If noisy correlations alone were to blame, the information objectives should catch up on the fully observed MMLU; yet they trail facility location significantly at $k = 5$ and 10 (Table 2).

Where facility location falls short. Facility location has two weak spots. Its first-step gain, one plus the sum of positive correlations, lets many weak correlations outweigh a few strong ones: on MERGED its first pick reaches $R^2 = 0.317$, against 0.490 for explained variance, a large gap that is nonetheless not significant (Table 2); with squared similarities, facility location makes the first pick of explained variance (Appendix G.6). And once most benchmarks are well covered, two candidates can tie exactly, for example two benchmarks that are each other's best proxy (Appendix H); the explained-variance gain, which uses $\hat{R}^{-1}_{\mathcal{A}\mathcal{A}}$, then breaks the tie, among the first ten picks in 24 of 50 splits on MERGED and 46 on MTEB, so late picks partly follow explained variance.

7 New models and real budgets

A selection is meant for models that appear after it is made, so we test whether it stays useful as models improve over time.

New models. We split three matrices by date: *old models*, which appeared before a cutoff, estimate the correlations and choose the benchmarks, and *new models*, which appeared after it, are predicted. HELM Lite goes from release v1.0 to later releases, MTEB from our snapshot to later models (MTEB-2026), and OLL v2 from submissions before 2025 to later ones, without its mathematics sub-tasks, whose scorer changed. Random splits of the same sizes give a control without a time shift; each method is scored on the new models that observe one of its picks, and differences are tested by bootstrapping over them (Appendix B.4).

With five benchmarks, facility location has the highest R^2 on the new HELM Lite and OLL v2 models and is close to the best on MTEB-2026, while MI falls behind on all three (Figure 4). On

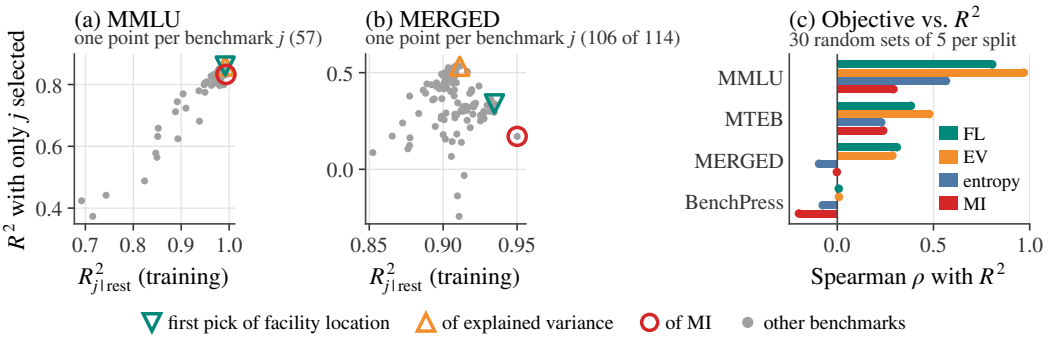


Figure 3: The information objectives choose poor proxies. (a, b) One point per benchmark: $R^2_{j|\text{rest}}$ under $\hat{R}$, which drives MI's first pick, against the realized R^2 when j alone is selected (means over the 50 splits); a good proxy sits high, and markers show each method's most frequent first pick (on MERGED, MI's lies far right but low; (b) omits 8 columns that no validation model has run when they can be selected). (c) Spearman correlation of each objective with realized R^2 over 30 random sets of five per split: the coverage objectives track the metric best, except on BenchPress, where none does.

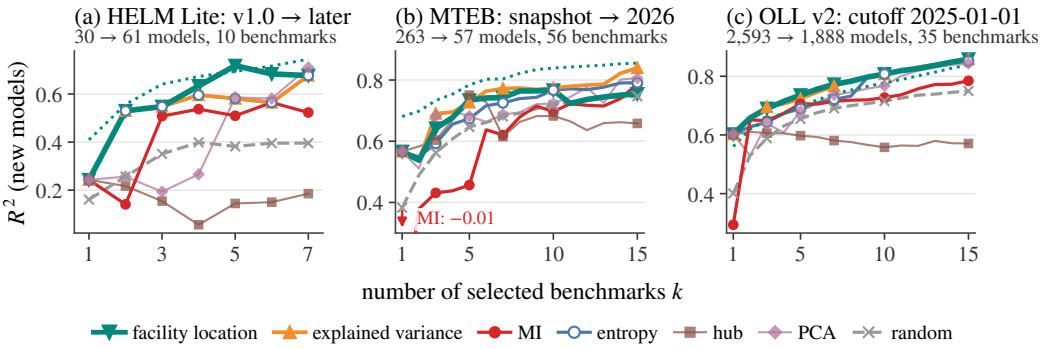


Figure 4: Facility location stays at or near the top on models released after the selection, and MI falls behind. R^2 on new models (one split per panel); dotted: facility location on random splits of the same sizes; more in Appendix E.1.

MTEB-2026 hub has the highest R^2, but only 24 of the 57 new models observe one of its picks, against 50 for facility location and explained variance. New models can also score higher or lower than old ones across the board, and (1), which measures errors against the old models' mean, partly credits a selection for tracking such a shift; against the new models' own mean, hub's lead disappears (Appendix E.1). Neither coverage objective dominates: with three benchmarks, explained variance is significantly ahead on OLL v2 and MTEB-2026, and with five at the two latest cutoffs of a fourth time split, Epoch, where new models score far above old ones (Appendix E.1).

Budgets. Benchmarks differ widely in cost (an MMLU subject has between 100 and 1,534 questions), so a fixed number of benchmarks is the wrong budget. Knapsack selection (Proposition 3b) prices each benchmark by its cost, which matters on MMLU: at 1% of the suite's questions, knapsack facility location reaches $R^2 = 0.836$, against 0.373 for cost-blind facility location (its greedy order, cut at the budget) and 0.707 for random (Figure 5 in Appendix F). Knapsack explained variance is significantly ahead of knapsack facility location at 6 of 7 MMLU budgets, by at most 0.006, and taking the cheapest subjects first is a strong baseline there (Appendix F.1); priced by cost, facility location thus keeps nearly all of what explained variance achieves, with a constant-factor guarantee that we cannot give for explained variance. For the same number of questions per model, whole benchmarks chosen by knapsack facility location predict all scores significantly better than questions drawn uniformly, from 800 questions per model on (OLL v1 items; Appendix F.4).

Further checks. Facility location and explained variance remain the two best methods on MMLU, MTEB and MERGED when each family of related models is held out in turn, though a held-out family is usually predicted less well (Appendix E.2); a selection made without a group of benchmarks, such as one type of MTEB task, predicts that group no better than random, since coverage extends only to the benchmarks one asks to cover (Appendix G.4). On the sparse matrices, subsampled training models and other correlation estimators leave facility location's R^2 unchanged, and a benchmark inflated by simulated contamination is never among the first five picks of a coverage objective (Appendices G.1 to G.3).

8 Related work

Fewer items or fewer benchmarks. Most work on cheaper evaluation keeps a few items (questions) per benchmark, chosen by model confidences [Anchor Points; Vivek et al., 2024], by item response theory [IRT; Maia Polo et al., 2024, Kipnis et al., 2025, Spagliardi et al., 2026], adaptively [Hofmann et al., 2025, Truong et al., 2025], or for reliability [Perlitz et al., 2024]; random items plus regression outperform most variants of the first two, which degrade when new models outperform old ones [Zhang et al., 2025]. Across benchmarks, scores are low-dimensional [Burnell et al., 2023, Ruan et al., 2024, Maia Polo et al., 2025], so low-rank completion predicts missing scores from a few probe benchmarks, which Zeng and Papailiopoulos [2026], in the paper behind BenchPress, choose greedily by the completion error of their imputer; other selectors cluster benchmark embeddings [Ye et al., 2023], enumerate predictive subsets [Surkov and Yamshchikov, 2024], or search for representative subsets [Subramani et al., 2025]. We instead derive the objective from the metric of a fixed predictor; for our linear predictor, the rule of Zeng and Papailiopoulos becomes greedy explained variance.

Facility location and experimental design. Closest to us, Yao et al. [2026] select prompts by facility location on text embeddings and, in our benchmark-level setting, found facility location on clipped score correlations to match or beat their reimplementation of our earlier entropy and MI for most k on MMLU and MTEB; we adopt their objective, relate it to the metric, compare it with explained variance on many more matrices, and explain why the information objectives fall behind. Measured on the residual covariance S, entropy and explained variance are the D- and A-optimality criteria of Bayesian experimental design [Pukelsheim, 2006]; greedy column subset selection maximizes a sample version of explained variance [Farahat et al., 2011], and sensor placement also uses variance reduction [Krause et al., 2008b,a].

9 Discussion and limitations

Benchmark scores are redundant enough that a few benchmarks predict the rest, and the objective that chooses them should follow from how the predictions are judged: coverage, not information. Explained variance is the metric on the estimated correlations; in cross-validation it never trails facility location significantly and leads it significantly only in a few dense settings, by a few thousandths of R^2, and on later models neither dominates. In exchange, facility location offers a data-independent guarantee, also under a budget, and readable selections, so we recommend it, with knapsack selection when costs differ. Selection pays most when few benchmarks fit the budget; with many, a random set is a reasonable fallback on dense matrices, and where the correlations cannot be estimated, the choice of selector matters little. For benchmark designers, a new benchmark adds information when the existing ones jointly predict it poorly, as Harfi et al. [2026] report for a new evaluation dimension.

Limitations. Our predictor is linear, though it needs no normality (Appendix B.6), and the correlation estimate assumes that scores are missing at random, which is unlikely, since which models run which benchmarks depends on age, provider and popularity. The guarantees bound the objective on the estimated correlations, not the accuracy of predictions; the evidence on models released later rests on one split per setting, unseen model families cost accuracy, and how fast a selection goes stale remains open. We did not model human-evaluation cost or test item selectors based on item response theory [Maia Polo et al., 2024]; contamination was simulated.

Broader impact. Pruning benchmarks could lead to overfitting on a narrowly selected set, rather than improving system performance. Seeding the selection with mandatory benchmarks, which keeps the facility-location gain monotone and submodular, and periodic full evaluations reduce these risks.

Acknowledgments and Disclosure of Funding

This work received no third-party funding. The author declares no competing interests.

References

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open LLM Leaderboard. `https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard`, 2023. Archived June 2024; results at `https://huggingface.co/datasets/open-llm-leaderboard-old/results`.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

Andrew An Bian, Joachim M. Buhmann, Andreas Krause, and Sebastian Tschiatschek. Guarantees for greedy maximization of non-submodular functions with applications. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 498–507, 2017.

Remco R. Bouckaert and Eibe Frank. Evaluating the replicability of significance tests for comparing learning algorithms. In Honghua Dai, Ramakrishnan Srikant, and Chengqi Zhang, editors, *Advances in Knowledge Discovery and Data Mining (PAKDD 2004)*, volume 3056 of *Lecture Notes in Computer Science*, pages 3–12. Springer, 2004. doi: 10.1007/978-3-540-24775-3_3.

Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1433–1452. SIAM, 2014. doi: 10.1137/1.9781611973402.106.

Ryan Burnell, Han Hao, Andrew R. A. Conway, and José Hernández-Orallo. Revealing the structure of language model capabilities. *arXiv preprint arXiv:2306.10062*, 2023.

Peter Businger and Gene H. Golub. Linear least squares solutions by Householder transformations. *Numerische Mathematik*, 7:269–276, 1965. doi: 10.1007/BF01436084.

Gerard Cornuejols, Marshall L Fisher, and George L Nemhauser. Exceptional paper-location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management science*, 23(8):789–810, 1977.

A. Das and D. Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML*, pages 1057–1064. Omnipress, 2011.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–38, 1977.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. In *Conference on Language Modeling*, 2024.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, London, 1993.

Epoch AI. Capabilities & Benchmarking. `https://epoch.ai/benchmarks`, 2026. Data snapshot `benchmark_data.zip` retrieved 17 October 2026; licensed CC BY 4.0.

Ahmed K. Farahat, Ali Ghodsi, and Mohamed S. Kamel. An efficient greedy method for unsupervised feature selection. In *Proceedings of the 11th IEEE International Conference on Data Mining*, pages 161–170, 2011. doi: 10.1109/ICDM.2011.22.

Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4): 634–652, 1998.

Jacob Feitelberg, Dwaipayan Saha, Kyuseong Choi, Zaid Ahmad, Anish Agarwal, and Raaz Dwivedi. TabImpute: Universal zero-shot imputation for tabular data. *arXiv preprint arXiv:2510.02625*, 2025.

C. Guestrin, A. Krause, and A. Singh. Near-optimal sensor placements in gaussian processes. In *International Conference on Machine Learning ICML'05*, 2005.

Sepehr Harfi, Ahmad Salimi, Dongming Shen, and Alex Smola. ProactBench: Beyond what the user asked for. *arXiv preprint arXiv:2605.09228*, 2026.

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

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

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

A. Krause and C. Guestrin. A note on the budgeted maximization of submodular functions. CMU-CALD 05-103, CMU, 2005.

Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(12), 2008a.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008b.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-Hard and BenchBuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024.

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*, 2023a.

Percy Liang, Yifan Mai, Josselin Somerville, Farzaan Kaiyom, Tony Lee, and Rishi Bommasani. HELM Lite: Lightweight and broad capabilities evaluation. `https://crfm.stanford.edu/2023/12/19/helm-lite.html`, 2023b. Stanford CRFM blog post, 19 December 2023.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. WildBench: Benchmarking LLMs with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*, 2024.

Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. Wiley, 3rd edition, 2019.

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*, volume 235 of *Proceedings of Machine Learning Research*, pages 34303–34326, 2024.

Felipe Maia Polo, Seamus Somerstep, Leshem Choshen, Yuekai Sun, and Mikhail Yurochkin. Sloth: Scaling laws for LLM skills to predict multi-benchmark performance across families. In *Advances in Neural Information Processing Systems*, volume 38, pages 39511–39556, 2025. doi: 10.52202/085713-1318.

Kanti V. Mardia. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57 (3):519–530, 1970.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

MTEB contributors. MTEB results repository. `https://github.com/embeddings-benchmark/results`, 2026. Snapshot commit 37dfa3b5 (13 March 2026); CC0-1.0 since 5 August 2026.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, 2023.

Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine Learning*, 52(3): 239–281, 2003. doi: 10.1023/A:1024068626366.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

Open LLM Leaderboard Team. Open LLM Leaderboard 2. `https://huggingface.co/collections/open-llm-leaderboard/open-llm-leaderboard-2`, 2024. Hugging Face collection.

Dimitris Papailiopoulos. BenchPress: Predicting LLM benchmark scores via low-rank matrix completion. `https://github.com/anadim/llm-benchmark-matrix`, 2026. Technical report and score matrix, February 2026.

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)*, pages 2519–2536, 2024.

Friedrich Pukelsheim. *Optimal Design of Experiments*, volume 50 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 2006. Reprint of the 1993 Wiley edition.

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*, 2024.

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

S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3–4):591–611, 1965.

Fabio Spagliardi, Mírian Silva, Ayan Datta, Aiden Zhou, Vamshi Krishna Bonagiri, and Diogo Cruz. Efficient safety benchmarking via item response theory. *arXiv preprint arXiv:2606.20626*, 2026.

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

Nishant Subramani, Alfredo Gomez, and Mona T. Diab. SimBA: Simplifying benchmark analysis using performance matrices alone. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13220–13233, 2025.

Maxim K. Surkov and Ivan P. Yamshchikov. Vygotsky distance: Measure for benchmark task similarity. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17409–17420, 2024.

SUSTech. mt_bench_judge. `https://huggingface.co/datasets/SUSTech/mt_bench_judge`, 2023. Hugging Face dataset, revision c489203c; 800 GPT-4 single-answer MT-Bench judgments of 5 models, copied from the LMSYS judgments at `https://huggingface.co/spaces/lmsys/mt-bench`.

Sang T. Truong, Yuheng Tu, Percy Liang, Bo Li, and Sanmi Koyejo. Reliable and efficient amortized model-based evaluation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 60238–60265, 2025.

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)*, pages 1576–1601, 2024.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh Agrawal, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. LiveBench: A challenging, contamination-limited LLM benchmark. In *International Conference on Learning Representations*, 2025.

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. Coresets before score sets: Evaluation-unsupervised prompt subset selection for LLM benchmarks. *arXiv preprint arXiv:2607.09739*, 2026.

Qinyuan Ye, Harvey Fu, Xiang Ren, and Robin Jia. How predictable are large language model capabilities? A case study on BIG-bench. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7493–7517, 2023.

Yuchen Zeng and Dimitris Papailiopoulos. You don't need to run every eval. *arXiv preprint arXiv:2606.24020*, 2026.

Guanhua Zhang, Florian E. Dorner, and Moritz Hardt. How benchmark prediction from fewer data misses the mark. In *Advances in Neural Information Processing Systems*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, 2023.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro von Werra. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, 2025.

A Proofs and guarantees

This appendix supports Section 3. Propositions 1 and 2 are simple facts about linear regression, and we include short proofs for completeness. Proposition 3 collects known guarantees; for each part we name its source, check its conditions for our objectives and tie rule, and report how close greedy comes to the optimum on our data. Throughout, R is a positive definite $N \times N$ matrix with unit diagonal and $k = |\mathcal{A}|$, with $0 < k < N$ in Proposition 1. For a nonempty set $\mathcal{A}$, $R^2(j \mid \mathcal{A}) = R_{j\mathcal{A}} R_{\mathcal{A}\mathcal{A}}^{-1} R_{\mathcal{A}j}$ is the share of the variance of benchmark j that a linear regression on the benchmarks in $\mathcal{A}$ explains. It equals 1 for $j \in \mathcal{A}$, and $R^2(j \mid \emptyset) = 0$.

Proof of Proposition 1. The identity is standard (cf. Das and Kempe, 2011, who maximize one of its summands, the R^2 of a single target). Let $z \in \mathbb{R}^N$ be the standardized scores of a new model, with mean 0 and covariance matrix R (unit variances), and let $W = R_{\bar{\mathcal{A}}\mathcal{A}} R_{\mathcal{A}\mathcal{A}}^{-1}$. The prediction error $e = z_{\bar{\mathcal{A}}} - W z_{\mathcal{A}}$ has covariance

$$\mathbb{E}[ee^\top] = R_{\bar{\mathcal{A}}\bar{\mathcal{A}}} - W R_{\mathcal{A}\bar{\mathcal{A}}} - R_{\bar{\mathcal{A}}\mathcal{A}} W^\top + W R_{\mathcal{A}\mathcal{A}} W^\top = R_{\bar{\mathcal{A}}\bar{\mathcal{A}}} - R_{\bar{\mathcal{A}}\mathcal{A}} R_{\mathcal{A}\mathcal{A}}^{-1} R_{\mathcal{A}\bar{\mathcal{A}}}.$$

Its diagonal entry for $j \in \bar{\mathcal{A}}$ is $1 - R^2(j \mid \mathcal{A})$, so $\mathbb{E}\|e\|^2 = \sum_{j \in \bar{\mathcal{A}}} (1 - R^2(j \mid \mathcal{A}))$. The denominator is $\mathbb{E}\|z_{\bar{\mathcal{A}}}\|^2 = \operatorname{tr} R_{\bar{\mathcal{A}}\bar{\mathcal{A}}} = N - k$. Hence

$$R^2_{\mathrm{pop}}(\mathcal{A}) = 1 - \frac{\mathbb{E}\|e\|^2}{\mathbb{E}\|z_{\bar{\mathcal{A}}}\|^2} = \frac{1}{N-k} \sum_{j \in \bar{\mathcal{A}}} R^2(j \mid \mathcal{A}) = \frac{F_{\mathrm{EV}}(\mathcal{A}) - k}{N - k},$$

where the last step uses $R^2(j \mid \mathcal{A}) = 1$ for the k selected benchmarks. The same computation shows that explained variance is N minus the total residual variance, $F_{\mathrm{EV}}(\mathcal{A}) = N - \operatorname{tr}(R_{\bar{\mathcal{A}}\bar{\mathcal{A}}} - R_{\bar{\mathcal{A}}\mathcal{A}} R_{\mathcal{A}\mathcal{A}}^{-1} R_{\mathcal{A}\bar{\mathcal{A}}})$. Only the mean and the covariance of z entered the argument. □

R^2_{pop} is the population version of the metric (1): with every cell observed, a predictor built from R and no ridge or clipping, the numerator and the denominator of (1), divided by the number of validation models, converge to $\mathbb{E}\|e\|^2$ and $\mathbb{E}\|z_{\bar{\mathcal{A}}}\|^2$. At fixed k, R^2_{pop} is an increasing affine function of F_{EV}, so each greedy step of explained variance on $\hat{R}$ maximizes the plug-in value of the metric; Appendix H compares that value with the realized R^2.

Proof of Proposition 2. The quantity $R^2(j \mid \mathcal{A})$ is the largest share of the variance of z_j that any linear combination of $z_{\mathcal{A}}$ explains: $R^2(j \mid \mathcal{A}) = 1 - \min_w \mathbb{E}(z_j - w^\top z_{\mathcal{A}})^2$. Restricting w to a single benchmark $i \in \mathcal{A}$ gives $\min_a \mathbb{E}(z_j - a\, z_i)^2 = 1 - R^2_{ij}$. Hence $R^2(j \mid \mathcal{A}) \geq \max_{i \in \mathcal{A}} R^2_{ij}$ for every j, and summing over j gives $F^{\mathrm{sq}}_{\mathrm{FL}}(\mathcal{A}) \leq F_{\mathrm{EV}}(\mathcal{A})$. For $\mathcal{A} = \{i\}$ the regression has a single regressor, so $R^2(j \mid \{i\}) = R^2_{ij}$ and the two sides agree; for $\mathcal{A} = \emptyset$ both vanish. □

In words, explained variance credits each benchmark with what the selected set explains jointly, and facility location with squared similarities credits it with what its best single proxy explains. The difference is what proxies add in combination. It vanishes whenever one proxy suffices.

Proposition 3(a): facility location. Part (a) is classical: Cornuejols et al. [1977] analyzed greedy selection for facility location, and Nemhauser et al. [1978] proved the factor $1 - 1/e$ for every normalized monotone submodular function. We repeat the short argument, which also covers the other similarities of Appendix G.6 and the restriction to eligible candidates. Fix a target benchmark j and let $c_j(\mathcal{A}) = \max_{i \in \mathcal{A}} s_{ij}$ be its cover, with similarities $s_{ij} \geq 0$ and $c_j(\emptyset) = 0$. Adding benchmark i raises the cover by $\max(s_{ij} - c_j(\mathcal{A}), 0)$. This gain is nonnegative, and it can only shrink as $\mathcal{A}$ grows, because $c_j(\mathcal{A})$ can only grow. Each c_j is therefore normalized, monotone and submodular, and so is their sum $F_{\mathrm{FL}} = \sum_j c_j$. This holds for $s = r^+$ and equally for the similarities R^2_{ij} and $|R_{ij}|$ of Appendix G.6, so greedy satisfies $F_{\mathrm{FL}}(\mathcal{A}_G) \geq (1 - 1/e) \max_{|\mathcal{A}| \leq k} F_{\mathrm{FL}}(\mathcal{A})$. No polynomial-time algorithm improves this factor for facility location with arbitrary nonnegative similarities, which contains maximum coverage, unless P = NP [Feige, 1998]. Eligibility restricts the candidates, not the targets. F_{FL} restricted to subsets of the eligible benchmarks is still monotone submodular, so the bound holds against the best eligible set.

Both coverage objectives count each selected benchmark once for itself ($c_j(\mathcal{A}) = 1$ and $R^2(j \mid \mathcal{A}) = 1$ for $j \in \mathcal{A}$), so $F \geq k$, and a factor on F promises less for the unselected benchmarks, $F(\mathcal{A}) - k$. The brute-force comparison under (c) is the practical check.

Tie-breaking keeps the guarantees. The greedy bound needs only that each step takes a largest gain, so any way of breaking exact ties keeps it, and a step that is optimal up to a small additive error loses only that error. Our rule picks among candidates within $\tau_t = 10^{-9}\max(g_t^*, 1)$ of the largest gain g_t^*, so the induction of Nemhauser et al. [1978] gives $F(\mathcal{A}_k) \geq (1 - 1/e)\,\mathrm{OPT} - \sum_t \tau_t$. Facility-location gains are at most N, so the loss is at most $kN \cdot 10^{-9} < 2 \times 10^{-6}$ on our matrices. The same argument covers the ratio ties of knapsack selection and the tied gains in Random Greedy.

Proposition 3(b): knapsack selection. Part (b) is the budgeted result of Krause and Guestrin [2005]; we describe the algorithm as we run it. Let $\kappa_j > 0$ be benchmark costs and C the budget. Cost-effective greedy starts from the empty set and repeatedly adds the eligible benchmark that still fits the remaining budget and has the largest gain per unit cost. It stops when no benchmark fits or no gain is positive. The best singleton is the affordable eligible benchmark with the largest $F(\{j\})$. Knapsack selection returns the singleton if it beats the greedy set G by more than $10^{-9}\max(1, F(G))$, and G otherwise. For a normalized monotone submodular F, the better of the two reaches at least $\frac{1}{2}(1 - 1/e) \approx 0.316$ times the value of the best affordable set; our tie rule lowers this bound by at most the 10^{-9} tolerances. The bound covers facility location only. Explained variance is not submodular, MI is not monotone and entropy needs a shift (below), so knapsack selection with these objectives is a heuristic. Cost-blind selection (Appendix B.5) has no guarantee, because an expensive early pick can use up most of the budget.

Proposition 3(c): explained variance. Part (c) combines a small counterexample with the data-dependent bound of Das and Kempe [2011]. F_{EV} is normalized and monotone, because a regression on more benchmarks never explains less. It is not submodular: a benchmark that explains little alone can explain more next to another one. Consider five benchmarks built from independent standard normal variables $u, v, \epsilon_3, \epsilon_4, \epsilon_5$:

$$z_1 = u, \qquad z_2 = \rho u + \sqrt{1 - \rho^2}\, v, \qquad z_i = \beta v + \sqrt{1 - \beta^2}\, \epsilon_i \quad (i = 3, 4, 5),$$

with $0 < \rho, \beta < 1$. Alone, benchmark 1 explains only benchmark 2, so its gain is $\Delta(1 \mid \emptyset) = 1 + \rho^2$. Next to benchmark 2 it lets the regression subtract u from z_2 and recover v, which predicts the three targets. Its gain becomes $\Delta(1 \mid \{2\}) = 1 - \rho^2 + 3\rho^2\beta^2$. The gain grows by $\rho^2(3\beta^2 - 2)$, which is positive for $\beta^2 > 2/3$.

Our data show the same effect. Along the greedy path of explained variance, 10% (MMLU) to 14% (BenchPress) of the candidates' gains grow after a pick, $\Delta(j \mid \mathcal{A}_t) > \Delta(j \mid \mathcal{A}_{t-1})$; each such case violates diminishing returns.

Without submodularity the guarantee depends on the data: greedy reaches $F_{\mathrm{EV}}(\mathcal{A}_G) \geq (1 - e^{-\gamma})\max_{|\mathcal{A}| \leq k} F_{\mathrm{EV}}(\mathcal{A})$, where the submodularity ratio

$$\gamma = \min_{L \subseteq \mathcal{A}_G,\ S \cap L = \emptyset,\ |S| \leq k} \frac{\sum_{x \in S}\big[F(L \cup \{x\}) - F(L)\big]}{F(L \cup S) - F(L)}$$

measures how far F is from submodular ($\gamma \leq 1$ always, with equality for submodular F). Das and Kempe [2011] state the bound for the R^2 of one target, but the proof uses only monotonicity and γ, so it carries over to the sum F_{EV} (see also Bian et al., 2017). We computed γ exactly at $k = 2$. Averaged over the 50 splits, it ranges from 0.71 (BenchPress) to 0.91 (MMLU), so at $k = 2$ the guarantee averages 0.51 to 0.60 of the optimum (as low as 0.40 on one BenchPress split). In practice greedy comes much closer. Against the brute-force optimum over all sets of size $k = 3$, and averaged over splits, greedy explained variance reaches at least 97.7% of the optimal value on each of the four main matrices (lowest on MTEB), and facility location at least 98.8% (lowest on MERGED; Table 12).

Proposition 3(d): mutual information. Part (d) restates known properties of MI, which we then measure on our data. The objective $F_{\mathrm{MI}}(\mathcal{A}) = I(z_{\mathcal{A}}; z_{\bar{\mathcal{A}}}) \geq 0$ is submodular [Krause et al., 2008b] and symmetric, $F_{\mathrm{MI}}(\mathcal{A}) = F_{\mathrm{MI}}(\bar{\mathcal{A}})$. In particular $F_{\mathrm{MI}}(\emptyset) = F_{\mathrm{MI}}([N]) = 0$, so unless it vanishes it must decrease somewhere. Two benchmarks with correlation 0.95 already show this: $F_{\mathrm{MI}}(\{1\}) = -\frac{1}{2}\log(1 - 0.95^2) \approx 1.164 > F_{\mathrm{MI}}(\{1, 2\}) = 0$. For non-monotone submodular functions, deterministic greedy has no constant-factor guarantee in general, and none is known for MI. Krause et al. [2008b] bound it up to an additive loss: if no gain on a set of at most $2k$ benchmarks falls below $-\epsilon$, their bound implies that greedy reaches at least $(1 - 1/e)\,\mathrm{OPT}_k - k\epsilon$. On MMLU and MTEB this loss is not small: the most negative gain of any candidate along greedy MI's path (on

sets of at most $2k$ benchmarks) puts $k\epsilon$ at no less than 16% and 29% of greedy's MI value at $k = 5$ (on MERGED and BenchPress no such gain was negative).

Along its own path, greedy MI stays in the increasing region. Its gain was positive at all 750 greedy steps (50 splits, 15 picks) on each of the four main matrices, and never below 0.10 nats (MMLU), whereas off the path some candidate had a negative gain in 100% of the MMLU splits and 100% of the MTEB splits. MI is monotone along the greedy path, not as a function.

Random Greedy restores a guarantee for every nonnegative submodular function: its set of at most k benchmarks satisfies $\mathbb{E}F(\mathcal{A}) \geq \mathrm{OPT}_k/e$ [Buchbinder et al., 2014]. We implement their Algorithm 1, which pads the candidates with $2k$ dummy benchmarks of zero gain and at each step draws uniformly among the k candidates with the largest gains. The guarantee holds for the k the algorithm ran with, not for prefixes of its output, so we ran $k \in \{5, 10, 15\}$ separately with five seeds per split. At $k = 5$ Random Greedy reaches a lower MI value than deterministic greedy on all four main matrices, by 0.24 to 0.36 nats on average, with every paired interval above zero. It does not predict better. Deterministic greedy is ahead on MMLU (0.906 against 0.902; difference +0.004 [+0.002, +0.006]), and on MTEB, MERGED and BenchPress the paired intervals include zero. The guarantee does not buy better predictions.

Proposition 3(e): entropy. Part (e) is a simple remark on a known objective. $F_{\mathrm{H}}(\mathcal{A}) = \log\det R_{\mathcal{A}\mathcal{A}}$ is submodular, because its gain $\log\sigma^2_{j|\mathcal{A}}$ can only shrink as $\mathcal{A}$ grows: conditioning on more benchmarks never increases a variance [Guestrin et al., 2005]. On a correlation matrix every conditional variance is at most 1, so every gain is at most 0. F_{H} never rises with a pick, and its first step is an exact tie. Over sets of at most k benchmarks its optimum is therefore the empty set, and the bound of Nemhauser et al. [1978], which needs a normalized monotone function, applies only to the shifted objective $F_{\mathrm{H}}(\mathcal{A}) + \delta|\mathcal{A}|$ for any $\delta \geq -\log\sigma^2_{\min}$, where $\sigma^2_{\min}$ is the smallest conditional variance of a benchmark given a set of fewer than $2k$ others. The shift does not change greedy's picks at fixed k, and the bound becomes $\log\det R_{\mathcal{A}_G\mathcal{A}_G} \geq (1 - 1/e)\,\mathrm{OPT}_{\mathrm{H}} - \delta k/e$ with $\mathrm{OPT}_{\mathrm{H}} = \max_{|\mathcal{A}|=k}\log\det R_{\mathcal{A}\mathcal{A}}$, the best set of exactly k benchmarks. Since δ grows without limit as a conditional variance approaches zero, and $\log\det R_{\mathcal{A}\mathcal{A}}$ is not the metric, this bound says nothing about prediction.

B Protocol details

B.1 Splits and random numbers

Repetition r of the cross-validation permutes the models with seed 42 ($r = 0$) or $(42, r)$ and cuts the permutation into 10 folds whose sizes differ by at most one. Split $s = 10r + f$ validates on fold f and trains on $\min\{\mathrm{round}((1 - p)M),\ \text{pool size}\}$ models drawn from the other nine folds, where the holdout fraction p is 0.1 in the main results. At $p = 0.1$ this is the whole pool, except where rounding leaves one pool model out (MERGED, Epoch, HELM MMLU and HELM Classic, in 10, 15, 5 and 15 of the 50 splits). At $p = 0.9$ (Appendix D) a random tenth of the models trains and the validation folds stay the same. Repetition 0 reproduces the folds of the submitted version. Every random stream is seeded by the split and by a tag of its own (`SEED_TAGS` in `cv_split.py`), never by the order in which methods run, so the random baseline draws the same permutation on a split in every experiment.

B.2 Estimating the correlation matrix

Each benchmark j is standardized with the mean and standard deviation of its n_j observed training scores ($n_j - 1$ in the denominator). The pairwise estimate sums the products of the standardized scores of two benchmarks over the n_{ij} training models that observe both, divides by $\max(n_{ij} - 1, 1)$, clamps the eigenvalues at 10^{-6} and rescales the result to unit diagonal.

The EM estimate [Dempster et al., 1977] treats the standardized scores z_m of the n_{tr} training models as draws from a Gaussian with mean μ and covariance Σ. For model m with observed benchmarks O_m and missing benchmarks U_m, the E-step and the M-step are

$$\hat{z}_{mU_m} = \mu_{U_m} + \Sigma_{U_mO_m}\Sigma^{-1}_{O_mO_m}(z_{mO_m} - \mu_{O_m}), \quad C_m = \Sigma_{U_mU_m} - \Sigma_{U_mO_m}\Sigma^{-1}_{O_mO_m}\Sigma_{O_mU_m},$$

$$\mu \leftarrow \frac{1}{n_{\mathrm{tr}}} \sum_m \tilde{z}_m, \qquad \Sigma \leftarrow \frac{1}{n_{\mathrm{tr}}} \sum_m \big[(\tilde{z}_m - \mu)(\tilde{z}_m - \mu)^\top + C_m\big],$$

where $\tilde{z}_m$ is row m completed by $\hat{z}_{mU_m}$ and C_m fills the $U_m \times U_m$ block. EM starts from the pairwise-complete covariance. After the start and after every M-step, eigenvalues are floored at ε, and an E-step adds a ridge of 10^{-6} to $\Sigma_{O_m O_m}$ if it is numerically singular. EM is regularized ($\varepsilon = 10^{-3}$, at most 1,000 iterations) if $n_{\mathrm{tr}} < N$ or fewer than half of the training cells are observed, and unregularized otherwise ($\varepsilon = 10^{-6}$, at most 500 iterations). It stops when $\|\Sigma_{t+1} - \Sigma_t\|_F / \|\Sigma_t\|_F < 5 \times 10^{-4}$. If $n_{\mathrm{tr}} < N$, the start and the result are shrunk as $\Sigma \leftarrow (1-\alpha)\Sigma + \alpha\,(\operatorname{tr}\Sigma/N)\,I$ with the fixed intensity $\alpha = (N - n_{\mathrm{tr}})/N$; this is the form of the estimator of Ledoit and Wolf [2004], not its optimal intensity. The final Σ is rescaled to a correlation matrix. At $p = 0.1$, $\hat{R}$ is pairwise on MMLU, OLL v2, OLL v1 items, HELM Lite and HELM MMLU; unregularized EM on MTEB, MTEB-2026 and HELM Classic; and regularized EM on MERGED ($\alpha = 0.07$), BenchPress and Epoch ($\alpha = 0$). Appendix G.3 shows how the results move when these settings change.

B.3 Selection and scoring rules

Eligibility. An ineligible benchmark stays in every objective: as a target of facility location and explained variance, in the complement of MI, in the hub scores and in the PCA loadings. It also stays an imputation target if at least two training models observe it. At $p = 0.1$ the eligibility rule binds on two matrices only. On MERGED it removes the eight MT-Bench columns, each observed for three models in total, in the 14 splits that leave one of these models out of training. On BenchPress it always removes AIME 2026 and GDP-Val AA, and it removes SciCode and MathVision in 14 and 3 splits.

Valid score ranges. Predictions are clipped to ranges that come from each metric's scale, never from observed scores: $[0, 1]$ for accuracies and other fractions (all of MMLU, the Open LLM, Epoch and HELM matrices, and most of MTEB and MERGED), $[-1, 1]$ for the Spearman correlations of the MTEB similarity and summarization tasks, $[0, 100]$ for percentages (win rates and pass rates in MERGED, most of BenchPress), $[1, 10]$ for MT-Bench, $[-80, 100]$ for WildBench scores, and no bound for Elo-style ratings in BenchPress. The clip acts in standardized units, so a prediction inside its range keeps its value; standardized inputs and true scores are also capped at ± 10.

Near-ties. A score counts as tied with the best score s^* if it lies within $10^{-9} \max(|s^*|, s_0)$ of it, where s_0 is the scale below which rounding is absolute: $s_0 = 1$ for the gains of facility location, explained variance and MI, for entropy (whose test is on the conditional variance $\sigma^2_{j|\mathcal{A}}$ rather than on its gain $\log \sigma^2_{j|\mathcal{A}}$) and for hub scores; the largest initial loading norm for PCA (below); and $s_0 = 0$ (a purely relative test) for gains per unit cost. Ties go to the larger explained-variance gain, then to the lower index. Explained variance, whose score is that gain, goes straight to the index; hub uses the gain at the empty set, so that its ranking stays one-shot; knapsack selection uses the gain per unit cost. The rule matters in four places: entropy's first step (an exact tie, which hands entropy the first pick of explained variance), MI's picks among the near-identical MT-Bench columns of MERGED, facility location's late picks (Appendix H) and equal-cost benchmarks in cheapest-first selection.

PCA subset selection. For each k, the Businger–Golub pivoted QR [Businger and Golub, 1965] runs on the rows of the top-k eigenvectors that belong to eligible benchmarks; each step pivots the benchmark whose loading vector has the largest norm after projecting out the earlier pivots. If k cuts through a repeated eigenvalue, the whole eigenspace is used, because its basis is arbitrary, and directions without loading on any eligible benchmark go last. PCA is the only selector whose sets are not nested in k.

B.4 Statistics

In the Nadeau–Bengio interval of Section 4, $n_{\mathrm{val}}/n_{\mathrm{tr}}$ is the ratio of the mean validation size to the mean training size over the splits used [Nadeau and Bengio, 2003, Bouckaert and Frank, 2004]. At $p = 0.9$ the ratio is about 1, so the interval is about 7.1 times as wide as the naive one. Standard deviations over splits describe spread and are not tests.

A split's R^2 is undefined when no validation model observes a selected benchmark. This happens only on sparse matrices at small k; for example, random is defined on 41 of the 50 BenchPress splits

at $k = 1$. Means skip undefined splits, and paired differences use the splits on which both methods are defined, so they can differ slightly from the difference of two means.

A prospective setting, which splits the models by release time (Section 7), has a single split. Its interval comes from 2,000 resamples of the new models with replacement, the same resamples for every method: we recompute each method's pooled R^2 on every resample and report the 95% percentile interval of the difference [Efron and Tibshirani, 1993]. Random enters as its mean over 50 draws within each resample. The interval covers the sampling of new models, not the variability of the selection.

B.5 Designs of the experiments on new models and costs

Prospective splits. A prospective split divides the models by release or submission date. Old models do everything that training models do in cross-validation: they fix the standardization, $\hat{R}$, the eligible set and every selection. New models are only imputed. New-model rows that duplicate an old-model row are dropped (identical observed values or, on OLL v2, identical model weights). On OLL v2 the seven MATH columns are dropped, because their scorer version changed during the leaderboard's life and is confounded with submission date; the "+ MATH" variant keeps them. On Epoch the benchmarks released on or after the cutoff are removed from both sides, so no score on a benchmark that did not exist at the cutoff enters the fit; the "by model" variant keeps them and therefore looks ahead. A new model enters a method's R^2 only if it observes one of the method's picks, so each method covers its own subset of the new models. The random-split control draws 50 random splits of the same models with the same training and test sizes; the prospective penalty is the prospective R^2 minus the control mean.

Model-family holdout. Families come from `model_families.py`: base-architecture keywords in model names for MMLU, MERGED, BenchPress and the two Open LLM matrices; the Hugging Face organization for MTEB and MTEB-2026; and the developer recorded in the metadata for Epoch and the HELM matrices. Fine-tunes go with their base model's family where the name or the metadata reveals it, and models without a family always stay in training. Every family with at least 40 (MMLU), 7 (MTEB, MERGED), 6 (BenchPress) or $\max(6, \lceil M/150 \rceil)$ models (other matrices) is held out in turn, and all other models train. Deterministic methods run once per family; random is averaged over five draws.

Budgets. Costs are the numbers of test questions of the MMLU subjects and the numbers of evaluation samples of the MTEB tasks (Appendix C). A budget is a fraction of the suite's total cost, the same on every split. Knapsack selection is described in Appendix A. Cost-blind selection fills the budget in the cardinality greedy order and skips benchmarks that no longer fit. Cheapest-first adds benchmarks by increasing cost. Random fills the budget from the split's permutation. Appendix F adds a fixed overhead per benchmark.

Whole benchmarks versus random items. The OLL v1 items matrix comes with all 28,659 item scores of its 395 models on 62 benchmarks. Each new model may answer Q questions. Whole-benchmark methods run knapsack selection with the number of items as cost and impute the rest. Random items take the first Q items of one uniform permutation of all items per split, shared by all validation models and nested across Q. Their estimator reads a benchmark's sample mean, centered by the training models' mean on the same items, as a noisy observation y_t of its standardized score, with a noise variance estimated from the training models' item residuals (with finite-population correction). It imputes all standardized scores by the Gaussian posterior mean $\hat{z} = \hat{R}_{\cdot o}(\hat{R}_{oo} + D + \lambda I)^{-1} y_o$, where o indexes the sampled benchmarks and D holds their noise variances. R^2 is pooled over all 62 benchmarks, and benchmarks run in full count exactly. Appendix F adds an overhead per benchmark touched.

B.6 Normality and determinism

The scores are not Gaussian. Shapiro–Wilk tests [Shapiro and Wilk, 1965] with a Benjamini–Hochberg correction at 5% [Benjamini and Hochberg, 1995] reject normality for 57 of 57 MMLU subjects and 56 of 56 MTEB tasks (median W 0.84 and 0.89), and for 60 of 114 MERGED benchmarks, where fewer observations per column give the test less power. Mardia's tests of multivariate

skewness and kurtosis [Mardia, 1970] also reject on MMLU and on the 133 complete rows of MTEB; MERGED has no complete row. Benchmark scores are bounded and often discrete. The Gaussian model is a working model: its conditional-mean formula is the best linear predictor for any zero-mean distribution with covariance R, and Proposition 1 uses only second moments.

Every job of the canonical run is single-threaded with a fixed hash seed in a pinned environment (Python 3.12, NumPy 2.4.6, SciPy 1.18.0), and two runs of the CPU pipeline produce byte-identical logs.

C Data, licenses and download defects

This appendix documents the score matrices of Section 4: where each comes from, under which terms we use and redistribute it, how MERGED is assembled, and two defects of our March 2026 downloads that were found after submission. Every matrix is built from public leaderboard results, and every build script, pinned revision and checksum is part of the release.

C.1 Matrices and their sources

Table 3 lists the eleven matrices of the paper, the pre-fix MERGED-v1 that serves only as a sensitivity check, the eight further collections that MERGED draws on, and the cost tables of the budget experiments (Section 7 and Appendix F). MMLU, MTEB, MERGED and BenchPress are the March/April 2026 snapshot used since the submission. The seven extended matrices were built in October 2026 from pinned inputs: each build script checks the SHA-256 checksum of every raw input and refuses inputs that differ. Epoch and HELM replace their files in place, so a rebuild from upstream succeeds only while the pinned versions remain available.

Costs. MMLU costs are test questions per subject, from 100 to 1,534. They sum to 14,042. MTEB costs are the `num_samples` of the evaluation splits that the matrix scores, from 100 to 8,848,803 per task, nearly five orders of magnitude. What one sample is depends on the task type: corpus documents plus queries for retrieval and reranking, texts for clustering and classification, sentence pairs for semantic similarity and pair classification, and source texts for summarization. The HELM token costs of Appendix F are HELM's own counts of prompt and output tokens per instance, times instances and requests per instance.

Further caveats. MT-Bench covers 5 models, so its 8 columns are observed for only 3 models of MERGED. BenchPress collects scores from heterogeneous public reports with different evaluation settings. In OLL v2, the MATH sub-tasks were scored by three versions of the scorer, so the split by submission date in Section 7 leaves MATH out. In OLL v1 items, the TruthfulQA items are continuous mc2 scores rather than 0/1. HELM Classic differs from the HELM website in 19 cells, where the website pools zero-shot ablation runs into the few-shot results. Epoch holds Epoch's normalized scores (the best variant per model group, rescaled between the random baseline and the ceiling), so one row may merge several versions of a model.

C.2 Licenses

We redistribute numeric scores only: model-by-benchmark matrices with observation masks, model and benchmark names, and metadata such as revisions, checksums, dates and evaluation costs. We redistribute no benchmark questions, reference answers, prompts or model-generated text. The one item-level file stores the per-item correctness of OLL v1 items models, indexed by position, without item text. The code is released under Apache-2.0. Each data file keeps the terms of its upstream source; we do not relicense upstream data. Our own additions (model-name mapping, metadata we authored) are released under CC BY 4.0. Attributions, our changes and the license texts or links to them are in `DATA_LICENSES.md` and `LICENSE-DATA` of the release; the TabImpute checkpoint we use is Apache-2.0. The letters below refer to Table 3.

(a) The dataset card declares no license, so use is governed by the Hugging Face Terms of Service, whose grant to other users covers use through the Hub. We redistribute only derived numeric scores, with attribution, and will remove files on request. Takedown requests go to `alex@smola.org`.

Table 3: Score matrices and their sources. *Revision*: the upstream commit, release or checksum that our file corresponds to; "equals rev. x": fetched without a pin, with all values equal to those at revision x. Letters refer to the license notes of Appendix C.2; $*$ marks the nine collections pooled in MERGED.

Matrix $M \times N$; observed	Source; revision	Retrieved	Upstream license (as declared)	We release
Main matrices				
MMLU*	Open LLM Leaderboard v1 [Beeching et al., 2023, Hendrycks et al., 2021], HF `open-llm-leaderboard-old/results`; 5-shot accuracy per subject; unchanged since 2024-07-18 (rev. 23474373)	2026-03-13	none declared [a]	matrix
MTEB 263 × 56; 77.3%	[Muennighoff et al., 2023, MTEB contributors, 2026], GitHub `embeddings-benchmark/results` at 37dfa3b5; MTEB(eng, classic)	2026-03-13	none then; CC0-1.0 since 2026-08-05	matrix
MERGED 118 × 114; 31.2%	the nine collections marked $*$, joined by canonical model names (Appendix C.4); rebuilt 2026-10-18 (Appendix C.3)	—	as its sources	matrix; name mapping
BenchPress 83 × 49; 33.8%	[Papailiopoulos, 2026], GitHub `anadim/llm-benchmark-matrix` at e93b33f1; scores from papers, model cards and leaderboards, each with a source URL	2026-04-10	README says MIT; no license file [e]	pinned download script; checksum
Extended matrices (pinned inputs with SHA-256 manifests)				
OLL v2 4,493 × 42; 100.0%	[Open LLM Leaderboard Team, 2024], HF `open-llm-leaderboard/results` at aa81ecc3 and `/contents` at 9c09a7ca; the 42 sub-task metrics behind the six v2 aggregates	2026-10-17	none declared [a]	matrix
OLL v1 items 395 × 62; 100.0%	tinyBenchmarks [Maia Polo et al., 2024] `lb.pickle` at e9a8b103: 28,659 items of 57 MMLU subjects and five further tasks, checked against the Open LLM Leaderboard v1	2026-10-17	MIT [f]	matrix; item correctness
Epoch 277 × 61; 17.7%	[Epoch AI, 2026], `epoch.ai/data/benchmark_data.zip`, pinned by SHA-256 (e1ce578c…); input table of the Epoch Capabilities Index	2026-10-17	CC BY 4.0; external sources keep theirs [g]	15 Epoch columns; build script
HELM Lite 91 × 10; 99.9%	[Liang et al., 2023a], bucket `crfm-helm-public`, release v1.13.0; objects pinned by storage generation and SHA-256; release dates from `stanford-crfm/helm` at 63754d05	2026-10-18	none for results; code Apache-2.0 [b]	matrix; token costs
HELM MMLU 79 × 57; 100.0%	as HELM Lite, release v1.13.0	2026-10-18	as above [b]	matrix; token costs
HELM Classic 67 × 16; 81.7%	as HELM Lite, release v0.4.0, few-shot runs only	2026-10-18	as above [b]	matrix; token costs
MTEB-2026 333 × 56; 72.4%	[MTEB contributors, 2026], HF `mteb/results` at 9d3f92b4; task and model metadata (release dates) of the `mteb` package 2.18.7	2026-10-18	CC0-1.0; metadata Apache-2.0	matrix
Other collections pooled in MERGED, and the pre-fix MERGED-v1				
Open LLM v2* 4,507×6	[Open LLM Leaderboard Team, 2024], the six aggregates; per model the result file that the leaderboard shows; rebuilt 2026-10-18 (Appendix C.3)	2026-10-17	none declared [a]	source table
HELM Lite v1.0* 30×11	[Liang et al., 2023b], bucket `crfm-helm-public`, release v1.0.0, `core_scenarios`	2026-03-13	as HELM Lite [b]	source table
AlpacaEval 2* 223×10	[Dubois et al., 2024], GitHub `tatsu-lab/alpaca_eval`, leaderboard CSV; equals rev. 2898342e	2026-03-13	code Apache-2.0; data badge CC BY-NC 4.0 [c]	source table
Arena-Hard-Auto* 60×4	[Li et al., 2024], GitHub `lmarena/arena-hard-auto`, leaderboard CSV of 2024-07-31; equals rev. b00effbe	2026-03-13	Apache-2.0	source table
LiveBench* 195×3	[White et al., 2025], HF `livebench/model_judgment`, mean judgment for the 3 of 6 categories it covers; rev. 9704e5da	2026-03-13	none declared [a]	source table
WildBench* 63×11	[Lin et al., 2024], HF Space `allenai/WildBench`, WB-Score per task category; equals rev. 4b94d1d8	2026-03-13	none declared [a]	source table
MT-Bench* 5×8	[Zheng et al., 2023, SUSTech, 2023], HF `SUSTech/mt_bench_judge`, GPT-4 scores averaged over both turns; equals rev. c489203c	2026-03-13	none declared; original "other" [d]	source table
BigCodeBench* 155×14	[Zhuo et al., 2025], HF `bigcode/bigcodebench-domain`, solve rate for 7 domains × {Complete, Instruct}; equals rev. 6176f481	2026-03-13	none declared [a]	source table
MERGED-v1 118 × 114; 31.1%	as MERGED, with the Open LLM v2 table of the submission	—	as its sources	matrix
Cost tables	test-split sizes of HF `cais/mmlu` at c30699e8 (metadata only); MTEB `num_samples` of the `mteb` package 2.18.7	2026-07-27; 2026-10-17	MIT; Apache-2.0	counts

(b) HELM's code is Apache-2.0, and HELM states no terms for its published results. We redistribute derived numeric scores with attribution to HELM [Liang et al., 2023a,b].
(c) The leaderboard CSV sits in an Apache-2.0 source tree, but the README's data badge states CC BY-NC 4.0. We treat the AlpacaEval scores, including the three AlpacaEval columns of MERGED, as CC BY-NC 4.0: attribution, non-commercial use, changes indicated.
(d) The 800 rows of the SUSTech dataset are copies of LMSYS's GPT-4 single-answer judgments for MT-Bench [Zheng et al., 2023]; the LMSYS Hugging Face Space declares "license: other" and states no terms. Note (a) applies.
(e) The README states MIT, but the repository has no license file and names no copyright holder. We therefore ship no BenchPress data, only a download script pinned to commit e93b33f1 that checks the file's SHA-256 checksum and rebuilds the matrix.
(f) The MIT license of tinyBenchmarks (© 2024 Felipe Maia Polo) covers `lb.pickle`, and its notice ships with OLL v1 items. The item results in it were produced by the Open LLM Leaderboard, whose per-model detail datasets declare no license.
(g) Epoch's data are free to use and redistribute with credit, but 46 of the 61 columns come from external projects that keep their own licenses. The release ships only the 15 columns taken from Epoch's own files and a script that rebuilds the full matrix from Epoch's download; it credits the creators of the other columns.
(h) BIG-bench Lite [Srivastava et al., 2023] (Apache-2.0) was downloaded but is used in no experiment: its models appear in no other collection. We include it in the release for completeness.

Personal information. Rows are identified by public model names. Hugging Face repository identifiers include the public user or organization name of the uploader; the matrices contain no other information about people.

C.3 Download defects

Two defects of our March 2026 downloads were found after submission. We fixed the first, which affected cells of MERGED, and disclose the second, which only drops models from MMLU.

Open LLM v2 and MERGED (fixed). Our original downloader kept each model's last result file in path order and, where a group entry was blank, stored the first sub-task's score as the group score. Rebuilding the Open LLM v2 table with the leaderboard's own choice of result file changes 200 cells in 47 of its 4,507 rows: 180 values and 20 cells that were missing. Two of these models are in MERGED, where the rebuild fills 8 missing cells, changes no observed value and raises the observed cells from 4,185 to 4,193. All results in this paper use the fixed matrix; the submission's matrix, MERGED-v1, serves as a sensitivity check (Table 4). The release includes the pre-fix files `open_llm_v1` and `merged_v1`, which reproduce the submission's numbers. The fix does not change the picture: facility location reaches $R^2 = 0.619$ at $k = 5$ on MERGED-v1 against 0.614 on MERGED, its lead over explained variance stays non-significant (FL − EV: +0.023 [−0.038, +0.084]), and its lead over MI stays significant (FL − MI: +0.133 [+0.037, +0.229]). Because MERGED-v1 is MERGED without these 8 cells, the scoreboard of Table 2 counts only MERGED.

MMLU (disclosed, kept as frozen). The MMLU downloader reads only the last results file of each model directory, so re-evaluated models show their last run, and 1,640 models whose MMLU results sit only in earlier files are missing. Writing to a case-insensitive file system also merged 18 model directories whose names differ only in case: 16 models were lost, and one row was filled from the run stored under the other casing (the two runs differ by up to 0.119). We keep the matrix as it was frozen for the submission and the rebuttal; of its 5,454 rows we drop the two that observe fewer than half of the subjects, which leaves 5,452. The defects decide which models and which run of a model enter the matrix; apart from that one row, every kept value is a score that the leaderboard published for that model. The downloader documents both effects and can recover the earlier files.

C.4 Construction of MERGED

MERGED pools the nine collections marked in Table 3, which share many models. MTEB stays out: its embedding models appear in no other collection. The same model often appears under different names (for example `Meta-Llama-3-70B-Instruct` and `llama-3-70b-instruct`). We matched

Table 4: Mean R^2 over the 50 matched splits at $k = 5$. (a) All eleven matrices with the holdout fraction $p = 0.1$ of the main results. FL − EV and FL − Random are paired differences with Nadeau–Bengio 95% intervals (Appendix B.4); an interval that excludes zero is significant. MERGED-v1 (last row) is MERGED before the data fix of Appendix C.3; it is a sensitivity check and not part of the scoreboard of Table 2. (b) Training on about a tenth of the models ($p = 0.9$) with the same validation folds; MI, hub and PCA were not run at this holdout fraction. Values for every k from 1 to 15 are in the released numbers file; Figure 2 plots them for the four main matrices.

(a) All matrices, $k = 5$, holdout $p = 0.1$

Matrix	FL	EV	MI	Entropy	Hub	PCA	Random	FL − EV [95% CI]	FL − Random [95% CI]
MMLU	0.919	**0.920**	0.906	0.913	0.875	0.907	0.890	−0.001 [−0.002, −0.001]	+0.028 [+0.020, +0.037]
MTEB	0.767	**0.790**	0.767	0.746	0.687	0.755	0.744	−0.023 [−0.089, +0.043]	+0.023 [−0.046, +0.092]
MERGED	**0.614**	0.593	0.465	0.503	0.481	0.483	0.563	+0.022 [−0.050, +0.093]	+0.051 [−0.025, +0.128]
BenchPress	0.053	−0.085	−0.059	−0.300	0.032	0.060	**0.120**	+0.138 [−0.292, +0.568]	−0.067 [−0.345, +0.211]
OLL v2	0.736	**0.738**	0.680	0.649	0.570	0.706	0.643	−0.002 [−0.003, −0.001]	+0.093 [+0.064, +0.123]
OLL v1 items	0.844	**0.845**	0.838	0.832	0.795	0.801	0.813	−0.001 [−0.006, +0.004]	+0.031 [+0.020, +0.042]
Epoch	**0.475**	0.458	0.183	0.426	0.416	0.303	0.306	+0.017 [−0.050, +0.084]	+0.169 [+0.028, +0.311]
HELM Lite	0.719	0.727	0.720	**0.729**	0.472	0.661	0.669	−0.008 [−0.036, +0.020]	+0.050 [−0.063, +0.163]
HELM MMLU	0.780	**0.786**	0.774	0.761	0.702	0.759	0.740	−0.006 [−0.030, +0.017]	+0.039 [+0.001, +0.078]
HELM Classic	**0.541**	0.352	0.450	0.508	0.417	0.339	0.418	+0.189 [−0.386, +0.764]	+0.123 [−0.217, +0.462]
MTEB-2026	**0.764**	0.752	0.752	0.720	0.651	0.716	0.710	+0.011 [−0.028, +0.050]	+0.054 [+0.024, +0.084]
MERGED-v1	**0.619**	0.596	0.486	0.514	0.466	0.471	0.564	+0.023 [−0.038, +0.084]	+0.056 [−0.017, +0.129]

(b) Fewer training models: $k = 5$, holdout $p = 0.9$

Matrix	FL	EV	Entropy	Random	FL − Random [95% CI]
MMLU	0.918	**0.919**	0.912	0.891	+0.027 [+0.004, +0.049]
MTEB	0.642	**0.653**	0.618	0.625	+0.017 [−0.111, +0.145]
MERGED	0.137	**0.146**	0.114	0.142	−0.005 [−0.189, +0.179]
BenchPress	0.095	**0.114**	0.105	0.106	−0.011 [−0.201, +0.179]
Epoch	**0.214**	0.199	0.166	0.176	+0.039 [−0.159, +0.236]
OLL v2	**0.735**	**0.735**	0.644	0.636	+0.099 [+0.011, +0.186]
MTEB-2026	0.645	**0.656**	0.639	0.623	+0.022 [−0.145, +0.188]

names across collections with the help of an LLM (Claude Sonnet 4.6; checklist Q16) and release the resulting mapping, which lists for every model of MERGED its original name in each collection where it appears. A model enters MERGED if it appears in at least two collections, which gives 118 models. Each column keeps its collection as a prefix (for example `mmlu/astronomy` or `helm/GSM8K`), so copies of one benchmark in two collections stay separate columns. We drop the ten auxiliary columns of AlpacaEval and Arena-Hard-Auto (standard errors, counts, interval bounds, response lengths and token counts), which are not benchmark scores. HELM Lite's "Mean win rate" stays: it is HELM's ranking statistic, which depends on the set of models, rather than a benchmark score, and we flag it here. The result has 114 columns, of which 31.2% of cells are observed.

D All eleven matrices and training-set size

Table 4 gives the numbers behind two statements of Section 5: panel (a) shows that the seven extended matrices follow the pattern of the four main ones, and panel (b) that with few training models the selector stops mattering on sparse matrices.

The extended matrices repeat the pattern. At $k = 5$ the best method is a coverage objective on every matrix except BenchPress (random) and HELM Lite, where entropy is ahead of explained variance by +0.002. On the three dense extended matrices (OLL v2, OLL v1 items and HELM MMLU), facility location and explained variance lie close together (FL − EV in the table) and above random and hub. On Epoch and MTEB-2026 facility location has the higher mean, though not significantly. The intervals against random include zero on MTEB, MERGED, BenchPress and the two small HELM matrices. Over every $2 \le k \le \min(15, N - 3)$, facility location is never significantly ahead of explained variance and is significantly behind in 22 of 144 settings, by at most 0.006, all on MMLU, OLL v2 and OLL v1 items.

Table 5: Prospective evaluation at $k = 5$: R^2 (1) on new models when old models estimate $\hat{R}$ and select. Old $\rightarrow$ new gives the numbers of models. Control: facility location's mean R^2 over 50 random splits of the same models with the same sizes. FL − EV: difference on the new models with a 95% percentile-bootstrap interval over new models (Appendix B.4); it omits the variability of the selection, and every setting is a single split. A new model counts for a method only if it observes one of the method's picks; "—" marks methods whose picks fewer than 10 new models observe.

Setting	Old → new	FL	EV	MI	Entropy	Hub	PCA	Random	Control	FL − EV [95% CI]
HELM Lite										
v1.0 → later	30 → 61	**0.718**	0.581	0.509	0.581	0.145	0.586	0.382	0.694	+0.137 [+0.066, +0.222]
MTEB-2026										
all new	263 → 57	0.734	0.727	0.457	0.674	**0.751**	0.681	0.647	0.784	+0.007 [−0.034, +0.032]
late releases	263 → 36	0.731	**0.764**	0.676	0.660	0.595	0.728	0.661	0.789	−0.033 [−0.073, +0.005]
OLL v2, cutoff										
2024-10	1,252 → 3,229	**0.717**	0.716	0.692	0.673	0.585	0.664	0.642	0.700	+0.001 [−0.003, +0.004]
2025-01	2,593 → 1,888	**0.736**	0.723	0.706	0.690	0.597	0.681	0.655	0.698	+0.012 [+0.007, +0.017]
2025-03	4,144 → 334	**0.765**	0.763	0.740	0.745	0.628	0.738	0.695	0.698	+0.002 [−0.005, +0.008]
2025-01, MATH	2,593 → 1,889	**0.779**	0.774	0.723	0.656	0.577	0.747	0.672	0.735	+0.005 [+0.001, +0.009]
Epoch, cutoff										
2024-07	83 → 194	0.882	0.286	0.329	**0.891**	0.148	0.375	0.588	0.365	+0.596 [+0.406, +0.828]
2025-01	119 → 158	0.635	0.682	0.627	**0.745**	0.635	0.661	0.511	0.409	−0.047 [−0.094, +0.003]
2025-07	164 → 113	0.462	**0.481**	0.278	0.374	0.243	0.202	0.341	0.417	−0.019 [−0.037, −0.004]
2026-01	194 → 83	0.441	**0.719**	0.359	0.356	0.290	—	0.410	0.408	−0.278 [−0.388, −0.071]
2025-01, model	119 → 158	0.633	0.658	0.518	0.642	**0.673**	0.366	0.512	0.366	−0.025 [−0.064, +0.009]

Few training models hurt only where correlations are hard to estimate. Trained on a random tenth of the models, facility location keeps $R^2 = 0.918$ on MMLU and 0.735 on OLL v2 (against 0.736 at $p = 0.1$), while on MERGED it drops to 0.137, level with random (0.142), and on BenchPress too every selector falls to the level of random. The Nadeau–Bengio correction alone makes the intervals of panel (b) about 2.8 times wider than those of panel (a), because it grows with the ratio of validation to training models.

E New models: prospective splits and model-family holdout

This appendix backs the results on new models in Section 7: *prospective* splits, in which old models choose the benchmarks and models released later are predicted, and the *model-family holdout*, in which each family of related models is held out in turn (designs in Appendix B.5). It adds further prospective settings, including Epoch, a correction for a shift of the new models as a group, and the family holdout on all eleven matrices.

E.1 Prospective splits

Settings. Besides the three settings of Section 7, Table 5 reports the OLL v2 cutoffs of October 2024 and March 2025, the OLL v2 variant that keeps the MATH columns (42 instead of 35 benchmarks), the 36 new MTEB-2026 models released after the snapshot (late), and Epoch. On Epoch models are split by release date at four cutoffs between July 2024 and January 2026.

More cutoffs, other k and missing scores. Facility location has the highest R^2 at all three OLL v2 cutoffs, and its lead over explained variance is significant only at January 2025 (Table 5). With the MATH columns, facility location is significantly ahead at that cutoff and explained variance at March 2025 (EV − FL: +0.0019 [+0.0003, +0.0036]; not in Table 5). With three benchmarks the order reverses: explained variance is significantly ahead on OLL v2 at January 2025 (EV − FL: +0.006 [+0.002, +0.010]) and on MTEB-2026 (+0.048 [+0.006, +0.108]), and on HELM Lite the two make the same picks at $k = 1$ and 3. The new MTEB-2026 models are mostly partial submissions: they observe 44% of their cells against 72% in the control splits, so their gap to the control mixes missing scores with drift.

Table 6: Model-family holdout at $k = 5$. (a) Each family with enough models is held out in turn; $\hat{R}$ is estimated and the benchmarks are selected without it, and the held-out family is imputed. Entries are unweighted means over families. Gap: facility location's family-holdout R^2 minus its cross-validation R^2 (Table 4); a negative gap is the cost of an unseen family. The gap compares an unweighted mean over families with the pooled R^2 of the matched splits. Jaccard: overlap between facility location's selection without the family and its selection on all models, averaged over families. (b) Facility location's R^2 on each held-out family of the four main matrices. There are no intervals, because the families are few and not exchangeable.

(a) All matrices, $k = 5$

Matrix	Families	FL	EV	MI	Entropy	Hub	PCA	Random	Gap	Jaccard
MMLU	8	0.899	**0.903**	0.890	0.897	0.852	0.881	0.866	−0.020	1.00
MTEB	10	**0.621**	0.620	0.572	0.560	0.500	0.542	0.550	−0.147	0.88
MERGED	9	**0.663**	0.619	0.464	0.492	0.596	0.504	0.562	+0.048	0.15
BenchPress	6	0.053	0.0002	−0.332	0.052	0.016	−0.085	**0.170**	+0.0003	0.14
OLL v2	9	**0.726**	0.717	0.637	0.592	0.515	0.644	0.607	−0.010	0.89
OLL v1 items	5	0.822	0.829	0.819	**0.837**	0.788	0.778	0.786	−0.023	0.87
Epoch	11	**0.369**	0.336	−0.141	0.342	0.260	0.259	0.196	−0.106	0.35
HELM Lite	6	0.382	0.412	**0.487**	0.412	0.327	0.355	0.471	−0.337	0.94
HELM MMLU	6	0.748	**0.753**	0.750	0.720	0.680	0.725	0.716	−0.031	0.64
HELM Classic	4	**0.683**	0.663	0.648	0.619	0.436	0.644	0.545	+0.141	0.58
MTEB-2026	15	0.602	**0.606**	0.542	0.520	0.424	0.489	0.541	−0.162	0.93

(b) Facility location's R^2 on each held-out family, $k = 5$

MMLU	gemma 0.896; gpt 0.957; llama 0.922; mistral 0.898; phi 0.855; qwen 0.917; solar 0.808; yi 0.935
MTEB	alibaba-nlp 0.801; baai 0.848; intfloat 0.493; jinaai 0.318; mcgill-nlp 0.597; mihaiii 0.641; minishlab 0.868; sentence-transformers 0.617; snowflake 0.411; voyageai 0.610
MERGED	claude 0.561; gemini 0.572; gemma 0.853; gpt 0.802; llama 0.673; mistral 0.675; phi 0.570; qwen 0.671; yi 0.587
BenchPress	claude −0.129; deepseek 0.180; gemini 0.301; gpt 0.121; llama −0.264; qwen 0.109

Correcting for a shift of the new models. New models can differ from old ones as a group. The R^2 of (1) measures errors against the training mean, so it also rewards a selection that merely tracks such a shift. Measuring against the new models' own mean removes this effect. Under this shift-adjusted R^2, facility location is first on HELM Lite (0.371 against 0.316 for explained variance), on MTEB-2026 (0.515 against 0.466 for explained variance and 0.225 for hub) and at every OLL v2 cutoff without MATH; only on the late MTEB-2026 models is facility location not first (explained variance 0.677, PCA 0.631, facility location 0.617). Hub's lead on MTEB-2026 thus disappears once the shift is removed. The adjustment also qualifies the HELM Lite result. Facility location's unadjusted R^2 is at the level of its control, but its shift-adjusted R^2 differs from the control's by -0.311, and every other method loses a similar amount: the new HELM Lite models are harder to predict than random ones, for every selector. These are point estimates without intervals.

Epoch. Epoch gives no stable ranking of the methods. New models there score far above old ones, so the unadjusted R^2 credits the shift: facility location's prospective R^2 exceeds its control by +0.517 at the July 2024 cutoff, while its shift-adjusted R^2 at the January 2025 cutoff is -0.963, and no method's is above zero there. The methods' picks also reach different numbers of new models: at the July 2024 cutoff the five benchmarks of facility location are observed for 65 new models and those of explained variance for only 12. The ranking of the methods changes with the cutoff: facility location is far ahead at July 2024, and explained variance significantly ahead at the two latest cutoffs (Table 5).

E.2 Model-family holdout

Holding out each model family in turn leaves facility location and explained variance the two best methods on MMLU, MTEB and MERGED (Table 6a), and on most matrices a held-out family is predicted less well than validation models in cross-validation (a negative gap). The gaps come with different changes of the selection. On MMLU the selection without a family is the one on all models

(Jaccard 1.00), and on MTEB and MTEB-2026 it barely changes (0.88 and 0.93), so their large gaps come from new models that differ from the training families, not from a different selection. On MERGED the selections change a lot (0.15) at no cost in R^2. On the other matrices the best method varies: facility location on Epoch, OLL v2 and HELM Classic, explained variance on HELM MMLU and MTEB-2026, entropy on OLL v1 items, random on BenchPress (as in cross-validation) and MI on HELM Lite. Families also differ among themselves (panel b): on MTEB facility location reaches 0.318 for the held-out jinaai models and 0.868 for minishlab. The MMLU gap understates the cost, because family labels come from model names and fine-tunes with uninformative names stay in training. The holdout is a split design: it tests transfer to unseen families, not the missing-at-random assumption of the estimator.

F Costs: budgets, fixed overheads, tokens and items

This appendix supports the budget paragraph of Section 7: it shows the budget curves, adds a fixed overhead per benchmark and token costs, and compares whole benchmarks with random items. All comparisons are paired over the 50 matched splits with Nadeau–Bengio 95% intervals (Appendix B.4).

F.1 Knapsack and cost-blind selection

Pricing benchmarks by cost matters on MMLU at small budgets, where cost-blind selection (the greedy order, filled to the budget) spends the budget on large subjects (Figure 5a). We compare knapsack selection (Proposition 3b) with cost-blind selection, cheapest-first and random at seven budgets, given as shares of the suite's cost. Knapsack explained variance is significantly ahead of knapsack facility location at 6 MMLU budgets, by at most 0.006. Cheapest-first is a strong baseline on MMLU, where a few directions dominate, so many cheap subjects predict nearly as well as a few chosen ones: it is significantly ahead of knapsack facility location at 3 budgets, from 2% to 5% (by up to 0.006), and significantly behind it at the other 4. On MTEB, knapsack and cost-blind facility location do not differ significantly, and knapsack facility location beats random significantly at only 2 budgets.

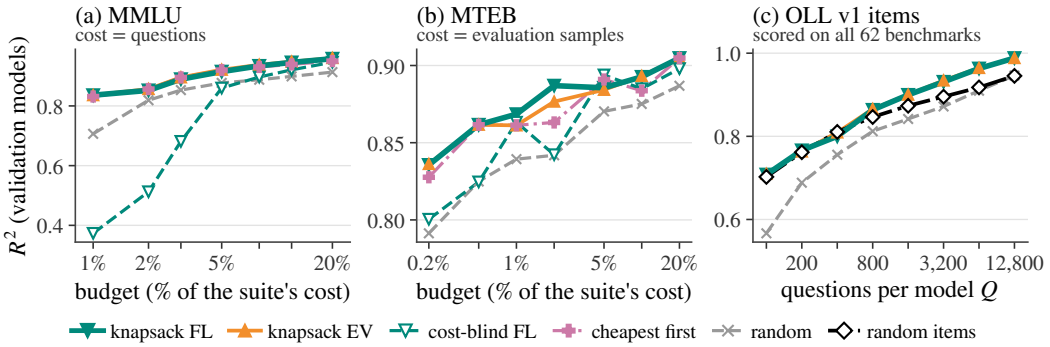


Figure 5: Pricing benchmarks by cost matters when the budget is small. (a, b) R^2 against the budget as a share of the suite's cost in questions (MMLU) or evaluation samples (MTEB), means over the 50 matched splits. Cost-blind selection spends small budgets on large subjects; knapsack selection fixes this. (c) OLL v1 items: R^2 over all benchmarks against questions per model; whole benchmarks (knapsack FL and EV, random) against Q uniformly random items.

F.2 A fixed overhead per benchmark

Running a benchmark has a fixed cost (integrating it into the harness, loading data and model, checking the output) on top of its per-question cost. We model it by adding a constant κ_0 to every benchmark's cost and set κ_0 to the median benchmark cost: 173 questions on MMLU and 32,078.5 evaluation samples on MTEB. Budgets are fractions of the suite's total cost including the overheads; otherwise the design is that of Figure 5. An overhead makes many small benchmarks expensive, which is exactly what cheapest-first buys.

Table 7: Knapsack facility location (FL) against five alternatives at every budget (share of the suite's cost), without and with a fixed overhead per benchmark (the median cost). ▲/▽: FL significantly ahead/behind (paired 95% Nadeau–Bengio interval); ·: no significant difference; –: no benchmark fits. Largest gap: FL minus the alternative at the budget where the difference is largest in absolute value.

	knapsack FL against	1%	2%	3%	5%	8%	12%	20%	largest gap
MMLU, no overhead	cheapest-first	▲	▽	▽	▽	▲	▲	▲	+0.008
	random	▲	·	▲	▲	▲	▲	▲	+0.129
	cost-blind FL	▲	▲	▲	▲	▲	▲	▲	+0.463
	knapsack MI	▲	▽	▽	▲	▲	▲	▲	+0.027
	knapsack EV	·	▽	▽	▽	▽	▽	▽	−0.006
MMLU, overhead	cheapest-first	–	▲	▲	▲	▽	▲	▲	+0.020
	random	–	▲	·	·	▲	▲	▲	+0.116
	cost-blind FL	–	▲	▲	▲	▲	▲	▲	+0.479
	knapsack MI	–	▲	▽	·	▲	▲	▲	+0.029
	knapsack EV	–	·	▲	·	▽	▽	▽	−0.005

	knapsack FL against	0.2%	0.5%	1%	2%	5%	10%	20%	largest gap
MTEB, no overhead	cheapest-first	·	·	·	▲	·	▲	·	+0.024
	random	·	▲	·	▲	·	·	·	+0.045
	cost-blind FL	·	·	·	·	·	·	·	+0.045
	knapsack MI	·	·	·	·	·	·	·	+0.009
	knapsack EV	·	·	·	·	·	·	·	+0.010
MTEB, overhead	cheapest-first	▲	·	·	▲	▲	▲	▲	+0.190
	random	·	·	▲	▲	▲	▲	▲	+0.112
	cost-blind FL	·	·	·	·	·	·	·	+0.082
	knapsack MI	▲	·	·	·	·	·	▲	+0.125
	knapsack EV	·	·	·	·	·	·	·	+0.005

With the overhead (Table 7), no benchmark fits the 1% budget of MMLU on any split (50 of 50), cheapest-first falls behind at all affordable MMLU budgets but one, and on MTEB knapsack facility location leads it by +0.190 [+0.105, +0.274] at 0.2%. The overhead also sharpens the case against random on MTEB. Knapsack explained variance is ahead of knapsack facility location at the three largest MMLU budgets with the overhead, and at no MTEB budget.

F.3 Tokens instead of evaluation samples on HELM

Section 7 measures cost in questions (MMLU) or evaluation samples (MTEB), a proxy for compute. For automatic evaluation the cost is closer to the number of tokens processed, which HELM reports per benchmark (Appendix C.1). On HELM MMLU the two agree well (Spearman correlation 0.83 over benchmarks), but not on HELM Lite (0.14) or HELM Classic (0.33).

We hold every selection to a true token budget and let knapsack facility location price benchmarks either by tokens or by their number of evaluation samples (HELM's instances). The metric is R^2 over all benchmarks of the suite, with the bought benchmarks counted as known; it compares selections of different sizes on the same targets. Table 8 shows that counts of evaluation samples are a proxy for token cost that can mislead: pricing in tokens helps more often than it hurts, but not at every budget. Priced in tokens, facility location beats random selection on HELM MMLU at every budget from 2%, on HELM Lite only at the large ones and on HELM Classic at none, and it differs significantly from explained variance priced in tokens only once.

F.4 Whole benchmarks versus random items

Section 7 compares whole benchmarks chosen by knapsack facility location with uniformly random items at a matched number Q of questions per model, on OLL v1 items (Figure 5c; design in Appendix B.5). Without overhead, whole benchmarks are significantly ahead at every budget from 800 questions on (at $Q = 12{,}800$: +0.043 [+0.037, +0.049]). We now charge $Q_0 \in \{10, 50, 100, 200\}$ extra questions for every benchmark that a model touches: once per selected benchmark, and once per

Table 8: Facility location (FL) priced by tokens on the three HELM suites, against three alternatives at every token budget (share of the suite's tokens); R^2 over all benchmarks, with bought benchmarks counted as known. Marks and the largest gap as in Table 7.

	FL priced by tokens against	1%	2%	5%	10%	20%	30%	50%	largest gap
HELM MMLU	FL priced by instances	·	▲	·	▲	·	·	·	+0.074
	random	·	▲	▲	▲	▲	▲	▲	+0.086
	EV priced by tokens	▽	·	·	·	·	·	·	−0.054
HELM Lite	FL priced by instances	·	·	▽	▲	▲	▲	·	+0.207
	random	·	·	·	·	▲	▲	▲	+0.101
	EV priced by tokens	·	·	·	·	·	·	·	+0.036
HELM Classic	FL priced by instances	·	▲	·	▽	▲	·	·	+0.205
	random	·	·	·	·	·	·	·	+0.081
	EV priced by tokens	·	·	·	·	·	·	·	+0.010

benchmark that the random items happen to fall into. With any of these overheads, whole benchmarks are significantly ahead from the smallest budget at which any benchmark fits on: 110, 150, 200 and 300 questions, the cost of the cheapest benchmark plus Q_0. This result is partly a property of the uniform item design: random items touch many benchmarks, so the overhead eats most of their budget, for example 82% at $Q_0 = 10$ and 110 questions, and 94% at $Q_0 = 50$ and 800 questions. An item design that knows the overhead, such as sampling a few benchmarks and many items within each, would fare better; we did not run one.

G Further checks

This appendix gives the design and the evidence behind the further checks of Section 7 (the model-family holdout is in Appendix E.2) and behind four choices of Sections 2 and 3: the Gaussian conditional mean rather than a fitted imputer, the similarity inside facility location, the clipping of predictions and the scale on which scores are modeled. Results are at $k = 5$ on the 50 matched splits unless stated otherwise; intervals are paired Nadeau–Bengio 95% intervals.

G.1 Selection stability

On sparse matrices the selections of different splits share few benchmarks. Low overlap could mean that selections are unreliable, and we test this directly. On every split we rerun each method on 10 random subsamples of 80% of the training models and score each of these alternative selections on the split's validation models, with the same imputer as the full selection. If low overlap were harmful, the alternatives would score below the full selection.

On MMLU the alternatives barely differ: their mean pairwise Jaccard similarity is 0.98 for facility location, and its full selections of different splits are identical (Jaccard 1.00), the same five subjects in all 50 splits. On MTEB the alternatives share about three of their five benchmarks (Jaccard 0.47) and score the same R^2 as the full selection: alternative minus full is +0.001 [−0.034, +0.037]. On MERGED they share few benchmarks (0.14; full selections of different splits 0.18), yet score no worse: +0.019 [−0.021, +0.060]. Explained variance behaves alike (0.10; +0.017 [−0.031, +0.065]). On BenchPress the overlap is lowest (0.09) and the difference, −0.004 [−0.223, +0.216], is too uncertain to say more. Low overlap on sparse matrices thus reflects many proxies of similar quality, not a selection that falls apart.

G.2 Contaminated benchmarks

A benchmark whose questions leaked into training data scores high for the affected models, which distorts its estimated correlations; we test whether the selectors then pick it. We simulate this on one target, MI's first pick (`miscellaneous` on MMLU, `BiorxivClusteringS2S` on MTEB and `arena_hard/score` on MERGED, there among benchmarks observed for at least 30 models): for a random 10% or 30% of the models we add 0.1 or 0.2 of the benchmark's score range to its score (capped at the maximum), re-estimate the correlations and rerun every selector on all models, with 20 random draws of the affected models for each of the four settings. In the strongest setting, 30%

Table 9: Sensitivity to the correlation estimate: R^2 at $k = 5$ when one setting of the estimator is changed and the rest of the pipeline is kept. Default: the paper's estimator, EM on both matrices, with no shrinkage on BenchPress (more training models than benchmarks) and shrinkage intensity $\alpha_0 = (N - n_{\text{tr}})/N$ toward the identity on MERGED. Pairwise (forced): pairwise-complete correlations instead of EM. α fixed: that shrinkage intensity on every split; $4\alpha_0$: four times the default intensity. EM tol.: convergence tolerance of EM (default $5 \cdot 10^{-4}$), the tighter one with four times as many iterations. PSD floor: smallest eigenvalue allowed during EM. Eligibility m: a benchmark is selectable if at least m training models observe it (default 3). Bold: the best method in each row.

Setting	FL	EV	MI	Random
BenchPress				
default	0.053	−0.085	−0.059	**0.120**
pairwise (forced)	0.291	**0.301**	0.059	0.230
$\alpha = 0.1$ fixed	0.105	0.124	0.107	**0.183**
$\alpha = 0.3$ fixed	0.153	0.182	0.152	**0.195**
EM tol. $5 \cdot 10^{-3}$	**0.210**	0.147	0.006	0.150
EM tol. $5 \cdot 10^{-5}$, 4× iter.	−0.137	−0.159	−0.042	**0.083**
eligibility $m = 10$	0.133	0.102	−0.058	**0.205**
MERGED				
default	**0.614**	0.593	0.465	0.563
pairwise (forced)	**0.619**	0.614	0.473	0.540
$\alpha = 0.3$ fixed	**0.583**	0.572	0.456	0.522
$\alpha = 4\alpha_0$	**0.585**	0.572	0.464	0.526
EM tol. $5 \cdot 10^{-3}$	0.632	**0.648**	0.565	0.571
PSD floor 10^{-6}	**0.622**	0.620	0.504	0.565
eligibility $m = 1$	**0.561**	0.538	0.465	0.511
eligibility $m = 2$	**0.561**	0.538	0.465	0.511

of the models and $+0.2$, facility location, explained variance and hub never place the target among their first five picks on any of the three matrices, in none of the draws; facility location moves it from place 3 to a median place of 14 on MMLU. Entropy places it among its first five in 90% of the MTEB draws and moves it from place 57 to a median place of 10 on MMLU: inflating some models' scores makes the target harder to predict from the other benchmarks, which is what entropy rewards. MI demotes its former first pick on MMLU and MTEB, but on MERGED it keeps the target among its first five in 5% of the draws in this setting and in 50% of the draws in the mildest one (10%, $+0.1$). The check is synthetic: it inflates one benchmark by a fixed amount and does not model which models leak which questions.

G.3 Correlation estimator

On sparse matrices the correlation matrix has to be estimated from incomplete data, and the estimator has several settings (Appendix B.2). We change one setting at a time and rerun selection and imputation on the matched splits; Table 9 shows the settings that matter on BenchPress and MERGED.

On BenchPress the estimate, not the selector, limits accuracy. Forcing pairwise correlations raises every method, random included: facility location from 0.053 to 0.291, explained variance from -0.085 to 0.301 and random from 0.120 to 0.230. Fixed shrinkage and a looser EM tolerance help as well, and a tighter tolerance hurts. Yet none of the 14 settings that change the estimate moves facility location significantly (the largest change, pairwise, is +0.238 [-0.040, +0.517]), and even with pairwise correlations facility location does not beat random significantly (+0.062 [-0.098, +0.222]). We therefore keep the default instead of choosing an estimator after seeing the results.

On MERGED none of the 16 settings changes facility location significantly. On MTEB every setting keeps it within 0.034 of the default, and none of these changes is significant. On MMLU, where every training row is complete and the default is the sample correlation, only fixed shrinkage matters, and it lowers R^2 ($\alpha = 0.3$: 0.882 against 0.919); a PSD floor of 10^{-2} changes R^2 by 0.0003.

Table 10: Held-out benchmark groups, $k = 5$: R^2 on the benchmarks of the held-out group, pooled over the held-out groups within a split and averaged over the 50 splits. Blind: selected without the group, from correlations estimated without it. Aware: selected outside the group by an objective that targets all benchmarks. Random: random benchmarks outside the group. The last two columns are paired differences with 95% intervals. Bold: the best R^2 in each row. Groups: MMLU and HELM MMLU, the four MMLU supercategories [Hendrycks et al., 2021] (STEM, humanities, social sciences, other); MTEB, the seven task types; MERGED, the source leaderboards, with all MMLU copies in one group and both MATH copies in another (10 groups, of which 9 enter the pool; MT-Bench, observed for 3 models, is left out); OLL v2, the six parent benchmarks; OLL v1 items, the four MMLU supercategories and each of the five other tasks.

	FL		EV	MI		paired differences, FL [95% CI]	
Matrix	blind	aware	blind	blind	Random	blind − Random	aware − blind
MMLU	0.901	0.902	**0.904**	0.896	0.886	+0.015 [+0.008, +0.021]	+0.0010 [+0.0001, +0.0018]
MTEB	0.668	**0.707**	0.689	0.680	0.676	−0.009 [−0.040, +0.023]	+0.040 [+0.004, +0.076]
MERGED	0.284	**0.367**	0.303	0.251	0.283	+0.001 [−0.180, +0.182]	+0.084 [−0.024, +0.191]
OLL v2	0.555	**0.611**	0.565	0.546	0.535	+0.020 [−0.017, +0.057]	+0.056 [+0.046, +0.065]
OLL v1 items	0.830	**0.839**	0.832	0.834	0.805	+0.025 [+0.014, +0.035]	+0.009 [+0.005, +0.013]
HELM MMLU	0.741	0.747	**0.755**	0.741	0.717	+0.023 [−0.017, +0.064]	+0.006 [−0.011, +0.023]

G.4 Held-out groups of benchmarks

When a suite gains a new kind of benchmark, a selection made from the old benchmarks has to predict it. We mimic this by holding out one group of related benchmarks at a time (groups in Table 10). The selection may not pick from the held-out group, and its benchmarks are scored by the R^2 of their imputation from the selection. A *blind* selection never sees the group: the selector runs on correlations estimated without it. An *aware* selection also picks outside the group, but its objective counts the group's benchmarks as targets, using the training models' scores on them.

A blind selection predicts the held-out group no better than random on MTEB (−0.009 [−0.040, +0.023]) and MERGED (+0.001 [−0.180, +0.182]). It is significantly ahead only on MMLU (+0.015) and OLL v1 items (+0.025), where one general factor links all groups. Making the objective aware of the group helps significantly on MTEB (+0.040 [+0.004, +0.076]), OLL v2 (+0.056), OLL v1 items and MMLU, and its point estimate is positive on every matrix. A coverage objective covers only the benchmarks it is asked to cover. Once some evaluated models have scores on a new kind of benchmark, it should enter the objective as a target, even if none of its benchmarks is selected.

G.5 Imputers

Section 2 keeps the Gaussian conditional mean because imputers fitted to the data did not predict significantly better from facility location's selections; here we also ask whether they favor a different selector. We score every selection under two matrix-completion imputers, SoftImpute (nuclear-norm regularized completion) and HardImpute (rank-r iterative SVD) [Mazumder et al., 2010], on the same cells as the Gaussian imputer (Table 11).

Neither fitted imputer beats the Gaussian conditional mean on MMLU, MTEB or MERGED. Relative to the Gaussian imputer, SoftImpute changes facility location's R^2 by −0.127, −0.076 and −0.166 on these three matrices, and HardImpute by −0.019, −0.051 and −0.050, the last two not significantly. On BenchPress SoftImpute is ahead by +0.126 [−0.102, +0.354], not significantly, and under it hub's selection does best (0.243). Under both fitted imputers, facility location or explained variance stays the best selector on MTEB and MERGED; under SoftImpute on MMLU, MI's selection does best (0.836 against 0.792 for facility location). The pretrained zero-shot imputer TabImpute [Feitelberg et al., 2025], run on the same splits and selections, falls far below the Gaussian conditional mean on MMLU, MTEB and MERGED (facility location: 0.539, 0.337 and 0.060 against 0.919, 0.767 and 0.614; on MMLU its context was subsampled to 3,800 models).

Table 11: Fitted imputers at fixed selections, $k = 5$. Each selection is scored by the Gaussian conditional mean of the paper and by SoftImpute and HardImpute, which are fitted to the training models together with the validation models' selected scores; all predictions are clipped to the valid range. The regularization of SoftImpute and the rank of HardImpute (at most k) are tuned on an inner holdout of training models. Last column: facility location's R^2 under the fitted imputer minus under the Gaussian imputer, with paired 95% interval. Bold: the best selector in each row.

Matrix	Imputer	FL	EV	MI	Hub	Random	FL: imputer − Gaussian [95% CI]
MMLU	Gaussian	0.919	**0.920**	0.906	0.875	0.890	—
	SoftImpute	0.792	0.788	**0.836**	0.809	0.786	−0.127 [−0.132, −0.121]
	HardImpute	**0.900**	**0.900**	0.892	0.859	0.866	−0.019 [−0.020, −0.018]
MTEB	Gaussian	0.767	**0.790**	0.767	0.687	0.744	—
	SoftImpute	**0.691**	0.671	0.657	0.631	0.614	−0.076 [−0.144, −0.008]
	HardImpute	0.716	**0.735**	0.684	0.666	0.670	−0.051 [−0.107, +0.004]
MERGED	Gaussian	**0.614**	0.593	0.465	0.481	0.563	—
	SoftImpute	**0.448**	0.428	0.302	0.384	0.389	−0.166 [−0.253, −0.078]
	HardImpute	**0.564**	0.514	0.271	0.301	0.398	−0.050 [−0.124, +0.023]
BenchPress	Gaussian	0.053	−0.085	−0.059	0.032	**0.120**	—
	SoftImpute	0.179	0.132	0.135	**0.243**	0.193	+0.126 [−0.102, +0.354]
	HardImpute	0.022	−0.144	−0.148	**0.170**	0.049	−0.031 [−0.233, +0.172]

G.6 Similarity in facility location

Facility location credits benchmark j with its best proxy's similarity, and the paper's similarity is the clipped correlation $r_{ij}^{+} = \max(R_{ij}, 0)$ of Yao et al. [2026], fixed before any experiment. Replacing it by $|R_{ij}|$ or by R_{ij}^2 changes no comparison significantly at $k \in \{1, 3, 5, 10, 15\}$ on any of the eleven matrices; over all $k = 1, \ldots, 15$ the only significant difference is on OLL v1 items at $k = 2$, where the clipped correlation is ahead of the squared one by +0.006 [+0.001, +0.012]. The squared similarity changes facility location's weak first pick on MERGED (Section 6): its score for a single benchmark, $1 + \sum_{i \neq j} R_{ij}^2$, is that of explained variance, so it makes the same first pick and reaches $R^2 = 0.490$ instead of 0.317 at $k = 1$ (clipped minus squared: -0.174 [-0.382, +0.035], not significant).

G.7 Clipping and logit space

Clipping to the valid range. Every prediction is clipped to its benchmark's valid score range (Section 2), and the clip is rarely active. On MMLU, MTEB and MERGED the largest share of predictions outside the range is 594 of 51,552 (MTEB, explained variance), and clipping changes R^2 by at most +0.005 (MTEB, entropy). On BenchPress the clip is active more often (up to 190 of 4,327 predictions) and matters more: it raises R^2 by +0.067 for facility location and by +0.193 for entropy.

Logit space. Fitting the Gaussian model to logit-transformed scores, which respects the bounds of accuracies, lowers the R^2 of facility location and explained variance on MTEB, MERGED and BenchPress (facility location on MERGED: 0.486 against 0.614) and leaves them level on MMLU; only the information objectives and random gain, and only on MMLU and BenchPress (entropy on BenchPress: -0.032 against -0.300), so we keep raw scores.

H Objective diagnostics

Table 12 collects the measurements behind the explanation of Section 6 and the guarantees of Proposition 3 on the four main matrices. Two effects need more than a table row: the late ties of facility location and the gap between the plug-in value of the metric and the realized R^2.

Rows 1–6 back Section 6, which discusses MMLU and MERGED; the table adds MTEB, where the correlations of row 2 lie between those of MMLU and MERGED, and BenchPress, where they are close to zero or negative. Three entries deserve comment. Hub's score ranks MMLU sets in reverse (-0.58): the most redundant sets have the highest hub score and predict worst. On MMLU the first pick of explained variance also draws most of its P_{jj} from the smallest eigenvalues (row 4); the share

Table 12: Objective diagnostics on the four main matrices, as means over the 50 matched splits. Rows 1–2: within-split Spearman correlation between an objective's value and the realized R^2, over all single benchmarks ($k = 1$) and over 30 random sets of five benchmarks per split. Row 3: $R^2_{j|\mathrm{rest}}$ of MI's first pick. Row 4: share of P_{jj} ($P = \hat{R}^{-1}$) that comes from the 10% smallest eigenvalues of $\hat{R}$, for the first picks of MI and EV. Row 5: mean $|\hat{R}_{ij}|$ among the five selected benchmarks. Row 6: the FL and EV sets at $k = 5$, as the share of shared picks and the Jaccard index. Row 7: greedy FL value relative to the brute-force optimum over all sets of size three (mean, worst split). Row 8: share of EV's candidate gains along its greedy path that grow after a pick, its submodularity ratio γ at $k = 2$ and the implied guarantee $1 - e^{-\gamma}$ (Appendix A). Row 9: realized minus plug-in R^2 at $k = 5$, in total and its observation-mask part (text). Row 10: FL's cover $F_{\mathrm{FL}}(\mathcal{A}_{10})/N$. Row 11: splits (of 50) with a tie-broken pick among FL's first ten.

	Quantity		MMLU	MTEB	MERGED	BenchPress
1	Spearman ρ, single benchmarks	FL	0.98	0.73	0.28	0.20
		EV	0.98	0.74	0.35	0.18
		MI	0.84	0.46	−0.03	−0.02
2	Spearman ρ, random sets of five	FL	0.81	0.38	0.31	0.01
		EV	0.97	0.48	0.29	0.01
		entropy	0.56	0.23	−0.09	−0.08
		MI	0.29	0.24	−0.002	−0.20
		hub	−0.58	−0.06	0.12	0.14
3	MI's first pick: $R^2_{j\|\mathrm{rest}}$		0.993	0.997	0.950	1.000
4	share of P_{jj} from the smallest 10% eigenvalues	MI first pick	95%	90%	98%	96%
		EV first pick	95%	54%	4%	69%
5	mean $\|\hat{R}_{ij}\|$ within the five picks	FL	0.71	0.74	0.40	0.42
		EV	0.70	0.68	0.35	0.42
		MI	0.89	0.70	0.28	0.39
		entropy	0.66	0.45	0.21	0.27
		hub	0.98	0.94	0.80	0.80
		PCA	0.63	0.45	0.22	0.29
		random	0.87	0.71	0.48	0.47
6	FL ∩ EV at $k = 5$	shared picks	80%	66%	21%	36%
		Jaccard	0.67	0.51	0.12	0.24
7	greedy / optimal F_{FL}, $k = 3$	mean	1.000	0.990	0.988	0.991
		worst split	1.000	0.987	0.961	0.968
8	EV: candidate gains that grow after a pick		10%	13%	13%	14%
	EV: γ at $k = 2$		0.91	0.82	0.76	0.71
	EV: bound $1 - e^{-\gamma}$		0.60	0.56	0.53	0.51
9	realized − plug-in R^2, $k = 5$	total, FL	−0.001	−0.079	−0.080	−0.666
		total, EV	−0.001	−0.059	−0.116	−0.842
		mask part, FL	−0.001	−0.013	−0.079	−0.127
		mask part, EV	−0.0004	−0.011	−0.103	−0.185
10	FL cover $F_{\mathrm{FL}}(\mathcal{A}_{10})/N$		0.969	0.938	0.824	0.813
11	splits with a tie-broken FL pick (first 10)		0	46	24	25

separates MI from explained variance only on the three sparser matrices. And greedy facility location reaches at least 96% of its optimal value on every split (row 7), far above its guarantee.

Late ties of facility location. After a few picks almost every benchmark has a close proxy. With ten picks the average cover $F_{\mathrm{FL}}(\mathcal{A}_{10})/N$ is 0.97 on MMLU and 0.82 on MERGED. The remaining gains are small, and they can tie exactly. Suppose two candidates i and j improve only their own cover and each other's. Their gains are $(1 - c_i) + (r^+_{ij} - c_j)$ and $(1 - c_j) + (r^+_{ij} - c_i)$, which are equal for any $\hat{R}$: facility location cannot choose between two mutual proxies. The tie rule of Appendix B.3 then picks the one with the larger explained-variance gain, which depends on $\hat{R}^{-1}_{\mathcal{A}\mathcal{A}}$. A tie-broken pick occurs among the first ten in 46 of the 50 splits on MTEB, 24 on MERGED and 25 on BenchPress, and in 0 on MMLU. Facility location's gains need only pairwise correlations, but its late picks on sparse matrices are partly decided by the explained-variance rule.

Plug-in versus realized R^2. Proposition 1 turns the objective of explained variance into a forecast of the metric: the plug-in value $R^2_{\text{pop}}(\mathcal{A})$ computed from $\hat{R}$. We split its error, realized minus plug-in R^2, into two parts. The mask part is the expected R^2 of the actual predictor (with ridge, on the cells and models the metric scores) if $\hat{R}$ were the truth, minus the plug-in value; it measures the loss from validation models that observe only some of the selected benchmarks. The rest, realized minus expected R^2, comes from estimation error in $\hat{R}$, from sampling the validation models and from clipping. For explained variance at $k = 5$, realized minus plug-in R^2 is -0.001 on MMLU: the plug-in value is nearly exact. On MERGED it is -0.116, almost all of it mask (-0.103): sparse validation rows observe few of the selected benchmarks. On BenchPress it is -0.842, of which only -0.185 is mask: the estimate itself is wrong, which is why we call this matrix estimation-limited (Sections 4 and 5). The same difference for facility location's sets is closer to zero on both sparse matrices (-0.080 on MERGED and -0.666 on BenchPress). Read off the greedy path, the forecast of Proposition 1 is accurate on dense matrices and optimistic on sparse ones.

I Changes since the submitted version

The submitted version recommended mutual information, evaluated selectors with ten cross-validation folds on three score matrices, and reported standard deviations over folds. Numbers in the public reviews and author responses therefore differ from the numbers here.

Recommendation and withdrawn claims. Facility location on clipped correlations, the objective of Yao et al. [2026], is the recommended default; the submitted version did not include it. Explained variance, which the submitted version only monitored, is now a main method, and we dropped a caveat about its stability that the data do not support. Entropy and MI remain as the original objectives, now with a negative result and its explanation (Section 6). We withdraw the claims that MI beats entropy at small k, the "surrogate gap" between them, and the recommendation of MI for small budgets. We also withdraw statements from our author response that the new runs contradict: that hub ranking finds the same first benchmark as MI; that cost-blind MI is the worst method at the tightest budget; that knapsack MI beats cheapest-first on MTEB; that MI "performs well throughout" and that objectives with guarantees degrade gracefully; and that temporal splits are infeasible.

What remains of the earlier claim. MI's lead over entropy at small k came from entropy's first step, where all benchmarks tie exactly and floating-point noise had picked the winner. With the tie broken by the explained-variance gain (Section 3), entropy makes the first pick of explained variance and beats MI on MMLU at $k = 1$ (entropy $-$ MI: +0.027 [+0.024, +0.030]; MI ahead in 0 of 50 splits). Beyond $k = 1$ neither is consistently ahead: over the 32 scoreboard settings of Table 2, MI is significantly ahead in 3 and entropy in 7.

Protocol, data and theory. Ten folds with ± 1 standard deviation bands became 50 matched splits shared by all methods and compared by paired Nadeau–Bengio intervals; the eligibility and target rules, the clipping of every imputer's predictions and the tie rule are new, and MTEB costs count only the evaluation samples of the splits that the matrix scores (Appendix B). The Open LLM v2 table and MERGED were rebuilt to fix a downloader defect, MMLU is kept as frozen with its caveats disclosed (Appendix C.3), BenchPress became a main matrix, and seven matrices were added, as were the experiments of Appendices E to G. Propositions 1 to 3 were added: the first two are simple facts, and the third collects known guarantees. MI's $1/e$ guarantee belongs to Random Greedy [Buchbinder et al., 2014], which we now run, not to deterministic greedy or to the randomized check of our author response; fitted imputers replace the submitted matrix-completion appendix (Appendix G.5), and the spectral lemma and the residual-trace diagnostic were dropped.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state four contributions: selection objectives derived from the prediction metric, and why entropy and MI do not follow from it (Sections 2 and 3); a paired comparison of seven selectors on eleven public score matrices, new models and real costs (Sections 4, 5 and 7); an explanation of why entropy and MI select poor proxies (Section 6); and a recommendation of facility location, with code and data. They claim no new theory. They also state the limits of these claims: facility location practically equals explained variance and is never significantly better than it; no selector beats random where correlations cannot be estimated well; on new models neither coverage objective dominates; and the evidence on new models rests on one split per setting.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Sections 4 and 9 list them: the linear working model (normality is rejected, although the conditional mean remains the best linear predictor), missingness at random as a working assumption, R^2 on sparse matrices measuring imputation from existing leaderboard coverage, guarantees that bound the objective rather than the accuracy on new models, prospective evidence from one split per setting, the accuracy cost of unseen model families, human-evaluation cost not modeled, the uniform item baseline, simulated contamination, and the snapshot defects disclosed in Appendix C.3.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Propositions 1 to 3 state their assumptions in Section 3; for example, Proposition 1 assumes full observation and the conditional mean without ridge or clipping. Propositions 1 and 2 are simple facts, proved in Appendix A for completeness; Proposition 3 collects known results, each cited with its conditions, which Appendix A checks for our objectives. Appendix A also shows that the tie rule keeps the guarantees and that, for MI, a constant-factor guarantee is known only for Random Greedy.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 and Appendix B specify the matched splits, the correlation estimator, the eligibility and target rules, the clipping of predictions, the ridge, the tie rule, the paired statistics and the design of every experiment; Appendix C gives every data source with its revision. The release reproduces all results with `run_all.sh`, which records software versions and data checksums in a manifest, and `run_tabimpute.sh` for the GPU experiment, both under pinned requirements.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

Justification: The repository at `https://github.com/smolix/benchmark-selection` contains the code (Apache-2.0), the data build scripts with pinned revisions and checksums, the score matrices where upstream terms allow (Appendix C.2), the logs, and the tool that generates every number of the paper from the logs. The entry points are `run_all.sh` (all CPU experiments) and `run_tabimpute.sh` (the GPU experiment).

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 and Appendix B give 5 repetitions $\times$ 10 folds with holdout fraction 0.1 and shared random numbers; pairwise correlations when at least 90% of the training cells are observed and there are at least as many training models as benchmarks, EM otherwise, with its tolerance, iteration cap, eigenvalue floor and shrinkage $\alpha = (N - n_{\mathrm{tr}})/N$; ridge $\lambda = 10^{-2}$; near-tie tolerance 10^{-9}; at least three training observations to be selectable and two to be scored; clipping to each benchmark's valid range. Budget grids, prospective cutoffs and family definitions are in Appendices B, E and G. The protocol rules were fixed before the final run (Appendix I); the fitted imputers of Appendix G.5 tune their regularization on an inner holdout of training models.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Methods are compared by paired differences over the 50 matched splits, with Nadeau–Bengio-corrected 95% t intervals (Section 4) that account for the overlap of training sets, and by win counts. "Significant" means that the interval excludes zero, per comparison and without correction for multiplicity, as Table 2 states. Results on new models use 2,000 bootstrap resamples of the new models, which omit the variability of the selection (Appendix B.4). Standard deviations over splits are descriptive. The variability comes from the partition of models, the subsampling in some designs and the random baselines.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The final run of all CPU experiments took 5,798 s of wall time on one server with an AMD EPYC 7502 (32 cores, 64 threads) and about 250 GB of memory, running 60 single-threaded jobs in parallel. The GPU experiment (the zero-shot imputer of Appendix G.5) ran for 23,497 s on one NVIDIA RTX 4090 (24 GB). Including preliminary, failed and rebuttal-era runs, the whole project used at most about 100 hours on RTX 4090 GPU servers, plus some laptop compute.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work analyzes public benchmark scores of language and embedding models. It involves no human subjects and no personal data beyond public model and uploader names, and it respects the terms of the data sources (Appendix C).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Selecting a few benchmarks reduces the compute and energy of evaluation. Section 9 names two risks: pruning a suite can leave rare but important capabilities, such as safety, under-evaluated, and a published selection can become an optimization target. It also names three mitigations: mandatory benchmarks can seed the selection, benchmarks whose imputed scores are uncertain can still be run, and periodic full evaluations refresh the correlations and expose gaming.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We release no models and no data with a high risk of misuse: the data are numeric scores that public leaderboards published.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Every source is cited, and Table 3 and the notes of Appendix C.2 give its revision, retrieval date and declared license or terms. Where a source declares no license we say so and redistribute only derived numeric scores, with attribution. Sources with restrictive or unclear terms are handled as stated there: the AlpacaEval scores keep their non-commercial CC BY-NC 4.0 terms, and BenchPress is not redistributed but fetched by a pinned, checksummed download script. The TabImpute checkpoint (Hugging Face `Tabimpute/TabImpute`, revision 72757498) is Apache-2.0, and the Python packages we use are cited or listed with their versions in the release.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the assembled score matrices where upstream terms allow, the model-name mapping of MERGED (Appendix C.4), the cost tables and all build scripts. They are documented by pinned revisions, SHA-256 manifests, metadata files and data-license files; Appendix C lists sources, licenses and known caveats (for example, the mixed MATH scorer versions of OLL v2). Upstream data keep their licenses; we do not relicense them. Our own additions (the model-name mapping and the metadata we authored) are released under CC BY 4.0.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work involves no crowdsourcing and no research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work involves no human subjects.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: An LLM (Claude Sonnet 4.6, Anthropic) helped match model names across leaderboards when we built MERGED (Appendix C.4); the released mapping lists every match. LLMs play no role in the selection, estimation or imputation methods.