BENCHMARK PREDICTION WITH SPARSE AND UNPAIRED CACHED RESPONSES

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating a generative model on all queries from all available benchmark tasks is prohibitively expensive. As such, models are often evaluated on different subsets of tasks and/or different subsets of queries from a given task. The resulting collection of responses and the corresponding (model, task) score matrix are sparse and unpaired. In this paper, we introduce the Product Kernel Perspective Space (PKPS), a collection of low-dimensional representations of models based on their embedded responses. PKPS generalizes existing methods for representing black-box generative models by comparing responses to similar – rather than just identical – queries and is therefore able to utilize sparse and unpaired information. Theoretically, we show that benchmark prediction using PKPS is query-efficient relative to methods that only use information from dense and paired portions of the evaluation data. Empirically, we show that PKPS outperforms baselines on benchmark prediction across two large benchmark suites.

1 INTRODUCTION

Evaluating a generative model is necessary for understanding its capabilities. As the capabilities of generative models increase (Wei et al., 2022; Bubeck et al., 2023), their applicability expands to more domains—often necessitating the creation of benchmarks for, e.g., mathematics (Hendrycks et al., 2021), law (Guha et al., 2023), finance (Islam et al., 2023), venture capital (Chen et al., 2025b), and agentic tasks (Liu et al., 2024). The current benchmarking landscape is correspondingly vast: BIG-bench alone contains more than 200 tasks (Srivastava et al., 2023), HELM tracks dozens of scenarios across multiple releases (Liang et al., 2023), and a recent compilation includes 133 different benchmarks for frontier models (Zeng and Papailiopoulos, 2026). As such, a model is rarely evaluated on all benchmarks: of the (model, task) pairs in the compilation above, only 23% have a reported score. Further, benchmarks evolve (i.e., queries are added (Chollet et al., 2025), refreshed to avoid contamination (White et al., 2025), or subsampled to reduce cost (Perlitz et al., 2024)) and models evaluated on the same benchmark may have responded to different sets of queries.

Benchmark prediction attempts to alleviate the resource burden of evaluation by using existing information rather than computing scores solely from newly generated responses. There are two main benchmark prediction tasks: query-efficient evaluation and (model, task) score completion. In query-efficient evaluation, a model responds to a small subset of a task's queries and its score on the full task is predicted from the resulting responses. Existing methods for query-efficient evaluation require access to item-level scores (Polo et al., 2024; Hofmann et al., 2025; Li et al., 2024; Bowyer et al., 2026) and/or cached embedded responses (Helm et al., 2026) to queries that all models have responded to and are typically not effective in settings where different models have been evaluated on different sets of queries. In (model, task) score completion, a model has not responded to any of a task's queries and its score on the task is predicted from the observed entries of the score matrix. Existing methods for score completion use only the aggregate scores and discard the item-level responses that produced them (Zeng and Papailiopoulos, 2026).

In this paper, we introduce the Product Kernel Perspective Space (PKPS) to predict benchmark scores directly from sparse and unpaired collections of responses. Our contributions can be summarized as follows:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: MAE of benchmark prediction methods on both suites and tasks (best per column bolded, with ties both bolded; second best underlined; setup described in Section 4): query-efficiency at budgets $m \in \{1, 8\}$ queries per cell at full task coverage, and completion of missing cells at cohort sizes $n \in \{10, \max\}$ with task coverage p_{task} and query depth p_{query} fixed at 0.5, except for the sp. columns where p_{task}=0.2 and n=10. "—" denotes a setting where a method is not applicable.

	HELM suite (18 tasks, 93 models)					**EEE suite** (16 tasks, 45 models)				
	Query-eff.		Completion			Query-eff.		Completion		
Method	(m=1)	(m=8)	(n=10)	(n=93)	(sp.)	(m=1)	(m=8)	(n=10)	(n=45)	(sp.)
Sample score	0.292	<u>0.092</u>	—	—	—	0.224	0.081	—	—	—
IRT	0.348	0.343	—	—	—	0.165	0.139	—	—	—
LRMC	—	—	0.139	0.105	0.208	—	—	0.109	0.096	0.144
DKPS	0.171	0.111	—	—	—	0.116	0.078	—	—	—
PKPS	<u>0.137</u>	0.109	**0.120**	<u>0.104</u>	**0.171**	<u>0.070</u>	<u>0.060</u>	**0.081**	<u>0.070</u>	**0.108**
Ensemble	**0.126**	**0.073**	**0.120**	**0.100**	<u>0.172</u>	**0.068**	**0.050**	**0.081**	**0.069**	**0.108**

1. *Conceptual:* we unify two prediction tasks as instances of a single problem: predicting scores from a sparse, unpaired collection of responses from previously evaluated models.
2. *Methodological & theoretical:* we propose the PKPS, a collection of low-dimensional representations of models defined on potentially unpaired embedded responses, and show that benchmark prediction using PKPS is at least as query-efficient as methods that only use information from paired responses (Theorem 2).
3. *Empirical:* we show that PKPS matches the accuracy of the sample score with up to an $8\times$ smaller query budget and, when ensembled with low-rank matrix completion, reduces score matrix completion error by up to 28%. Our empirical evaluation spans two large benchmark suites.

1.1 Related Work

Benchmark prediction. We decompose benchmark prediction into two lines of prior work: query-efficient benchmarking and (model, task) score completion. Methods for query-efficient benchmarking predict a model's score on a full task from its responses to a small subset of the task's queries—typically by selecting an informative query subset and extrapolating via item-response theory (Polo et al., 2024; Hofmann et al., 2025), clustering (Vivek et al., 2024), rank-preserving subsampling (Perlitz et al., 2024), learned acquisition policies (Li et al., 2024), regression on item-level scores (Bowyer et al., 2026), or regression on model representations (Helm et al., 2026). All previous methods proposed for query-efficient benchmarking implicitly require item-level scores or responses on a common set of queries across reference models.

Methods for (model, task) score completion predict entries of the score matrix that were never evaluated via cross-task predictability (Ye et al., 2023), observational scaling laws (Ruan et al., 2024), or low-rank matrix completion (Zeng and Papailiopoulos, 2026). All operate exclusively on aggregate scores and discard information in the item-level scores or responses. Our proposed method uses item-level information from sparse and unpaired cross-task evaluation data to improve on methods from both literatures.

Low-dimensional representations of models. Our approach to sparse and unpaired benchmark prediction first constructs low-dimensional representations of the models before learning a simple regressor in the representation space. Existing approaches to represent models vary by access regime. For example, representations can be induced by comparing internal activations (Duderstadt et al., 2023; Huh et al., 2024) and/or weights (Chen et al., 2025a) in the white-box setting. Evaluators do not necessarily have white-box access and instead rely on black-box methods such as the Data Kernel Perspective Space (DKPS) (Helm et al., 2024; 2025), which constructs low-dimensional representations based on distances between matrices of embedded responses on a paired set of queries. Most relevant to our setting, Helm et al. (2026) regress benchmark scores of previously evaluated models on DKPS representations and prove that the predictions are query-efficient relative to the sample score. The proposed Product Kernel Perspective Space generalizes this construction with a kernel on the query space, enabling comparisons in the sparse and unpaired regime.

2 BENCHMARK PREDICTION

For our purposes, a model is a random mapping $f : \mathcal{Q} \to \mathcal{X}$ from a query space to a response space. A task is a pair $(Q^{(t)}, s^{(t)})$, where $Q^{(t)} \subseteq \mathcal{Q}$ is a collection of queries with $M^{(t)} := |Q^{(t)}|$ and $s^{(t)}$ is a scoring function mapping a model and a query collection to a nonnegative real number. Model i's true score on task t is $y_i^{(t)} := s^{(t)}(f_i, Q^{(t)}) \in \mathbb{R}_{\geq 0}$.

Evaluation data is indexed by (model, task, query) triples: for n models $f_1, \ldots, f_n$ and T tasks $Q^{(1)}, \ldots, Q^{(T)}$, the response of model i to query $q \in Q^{(t)}$ may or may not have been generated and scored. Let $Q_i^{(t)} \subseteq Q^{(t)}$ be the queries for which model i's response (and score) is observed, with $m_i^{(t)} := |Q_i^{(t)}|$ and $Q_i := \cup_t Q_i^{(t)}$. The corresponding sample score $\tilde{y}_i^{(t)} := s^{(t)}(f_i, Q_i^{(t)})$ is an estimate of $y_i^{(t)}$.

Evaluation data is *sparse* when few models have $m_i^{(t)} > 0$ for a given task. It is *unpaired* when the sizes of cross-model query set intersections $m_{ii'}^{(t)} := |Q_i^{(t)} \cap Q_{i'}^{(t)}|$ are small. We let $\rho_{ii'}^{(t)} := m_{ii'}^{(t)}/m_i^{(t)}$ be the ratio of shared queries for a given model pair and note that $\rho_{ii'}^{(t)} \approx 1$ for all pairs in highly curated leaderboards and $\rho_{ii'}^{(t)} \approx 0$ in deployed evaluation settings. The goal of benchmark prediction is to estimate a model's score from the available evaluation data.

The two tasks introduced in §1 are the two regimes of the observation count $m_i^{(t)}$:

1. *Query-efficient evaluation* is the case with $m_i^{(t)} > 0$. Responses or scores from other models on task t are typically available.
2. *(Model, task) score matrix completion* is the case with $m_i^{(t)} = 0$. No responses from the target (model, task) slice exist and prediction must be based on model i's responses to other tasks and on responses from other models evaluated on task t.

3 THE PRODUCT KERNEL PERSPECTIVE SPACE

We estimate $y_i^{(t)}$ from sparse and unpaired evaluation data by constructing low-dimensional representations of the models based on an appropriately defined pairwise distance matrix. We assume access to the responses for each observed (model, task, query) triple and to two embedding functions g_Q and g_R that map queries and responses to $\mathbb{R}^{p_Q}$ and $\mathbb{R}^{p_R}$, respectively. Let $q_1, q_2, \ldots, q_M$ enumerate the observed queries and let x_{ij} denote the embedded response $f_i(q_j)$ and u_j denote the embedded query q_j.

Given a query kernel $k_Q : \mathbb{R}^{p_Q} \times \mathbb{R}^{p_Q} \to \mathbb{R}_{\geq 0}$ and a response kernel $k_R : \mathbb{R}^{p_R} \times \mathbb{R}^{p_R} \to \mathbb{R}_{\geq 0}$, we compare models i and i' by computing the pairwise affinity matrix A:

$$A_{ii'} = \frac{\sum_{j \in Q_i} \sum_{l \in Q_{i'}} k_Q(u_j, u_l)\, k_R(x_{ij}, x_{i'l})}{\sum_{j \in Q_i} \sum_{l \in Q_{i'}} k_Q(u_j, u_l)}. \tag{1}$$

Each entry is normalized by its own query-kernel mass, so models that answered different numbers of queries, or entirely different queries, remain comparable. We use the induced squared distance $D_{ii'}^2 = A_{ii} + A_{i'i'} - 2A_{ii'}$ to capture model (dis)similarity (Gretton et al., 2012). Finally, the Product Kernel Perspective Space representations of the models—$\widehat{\psi}_1, \ldots, \widehat{\psi}_n$—are defined as the d-dimensional Euclidean configuration that minimizes the raw stress criterion to D (Trosset and Priebe, 2024). That is,

$$(\hat{\psi}_1, \ldots, \hat{\psi}_n) = \underset{z_1, \ldots, z_n \in \mathbb{R}^d}{\arg\min} \sum_{i,i'=1}^{n} \left(\|z_i - z_{i'}\| - D_{ii'} \right)^2. \tag{2}$$

We note that the definition of the PKPS strictly generalizes previous work by replacing the identity query kernel ($k_Q(u, u') = \mathbf{1}\{u = u'\}$) with a generic query kernel. Importantly, for any strictly positive k_Q, such as RBF, $D_{ii'}$ includes information from unpaired responses and Eq. (1) is therefore always defined when $|Q_i|, |Q_{i'}| \geq 1$. We highlight the difference between the previous approaches and the PKPS in Figure 1. Finally, Eq. (2) is solved by classical multi-dimensional scaling (Torgerson, 1952; Borg and Groenen, 2005) whenever D is a Euclidean distance matrix (Dokmanić et al., 2015). Unless otherwise stated we analyze and apply PKPS under this assumption.

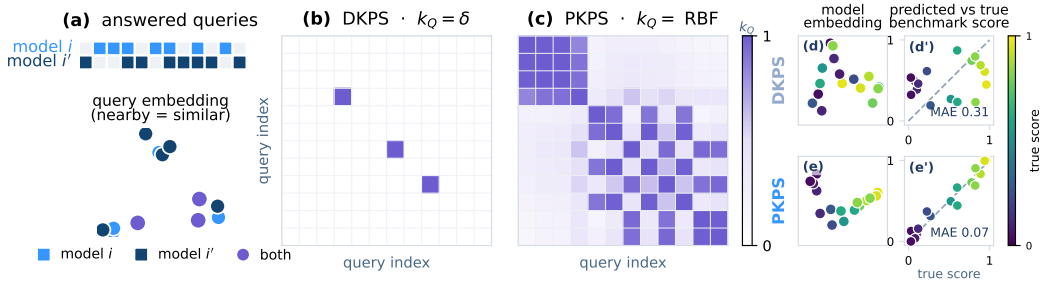

Figure 1: An illustration of the difference between the Product Kernel Perspective Space (PKPS) and the Data Kernel Perspective Space (DKPS). **(a)** Top: two models respond to different sets of queries. Bottom: queries represented as vectors, colored by which model has responded to them. **(b)** The query kernel used by DKPS. Only information from queries both models have responded to is included. **(c)** The query kernel used by PKPS weighs query similarity in the embedding space. **(d–e′)** A benchmark prediction example. Left: each dot is a model colored by its benchmark score. Right: true benchmark score versus predicted benchmark score. Restricted to the rare shared queries, DKPS (top) is noisy and correlates poorly with benchmark score (MAE 0.31). PKPS (bottom) leverages unpaired queries and the is highly correlated with true score (MAE 0.07).

3.1 Concentration of PKPS representations

We next establish theoretical properties of the PKPS necessary for demonstrating its efficiency in sparse and unpaired benchmark prediction. The main result of this section establishes that the representations concentrate around their population counterparts at a rate governed by the overlap of the query distributions. We model the evaluation data of Section 2 as sampled (task, query, response) triples: conditional on model i, each triple independently draws a task, then a query from the task, and finally the model's response (formalized in Definition 2 of Appendix D). We formally define the population PKPS in Definition 2 of Appendix D; broadly, it is defined by replacing the sums of Eq. (1) with expectations.

Our concentration result requires bounded query and response kernels, bounded embedded responses, a positive cross-model query kernel mass, and a rank-d spectral gap in the population Gram matrix (see Assumptions 2–4, Appendix D). The required boundedness and the spectral gap are standard in spectral-embedding analyses (Yu et al., 2015; Acharyya et al., 2024; 2025). The atypical requirement in our setting is the positive cross-model query kernel mass, though this holds trivially under a strictly positive k_Q and does not assume that any individual query is shared.

Let $(\psi_1, \ldots, \psi_n)$ denote the population PKPS, $\kappa = \inf_{i,i'} \mathbb{E}[k_Q(u, u')]$ the cross-model query kernel mass, and λ_d the d-th largest eigenvalue of the population Gram matrix.

Theorem 1 (Concentration of PKPS representations)**.** *Suppose Assumptions 2–4 hold and that every model has responded to at least m queries. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ there is an orthogonal W such that* $\max_i \|\hat{\psi}_i - W\psi_i\| \leq C\kappa^{-2}\lambda_d^{-1/2}\sqrt{d}\, n\sqrt{\log(4n^2/\delta)/2m}$, *where C depends only on the kernel bounds.*

Theorem 1 states that the PKPS representations concentrate at the rate in m enjoyed by fully paired DKPS (Acharyya et al., 2025) without requiring any shared queries. The price of unpairedness is a factor of κ^{-2}, governed by the overlap of the query distributions. The full proof is in Appendix D.

3.2 Inference in the PKPS

We next describe benchmark prediction in the PKPS and establish its query efficiency. Inference using the PKPS is a two-step statistical problem: first embed the models (Eq. (2)), then train a decision function that uses the representations as proxies for the models.

Fix a task t and let $f_0, f_1, \ldots, f_n$ be i.i.d. draws from a model distribution. Our goal is to predict the true score of f_0 on task t, $y_0 := s^{(t)}(f_0, Q^{(t)})$. The (model, task) pairs with $m_i^{(t)} > 0$ and model f_0's responses from tasks $t' \neq t$ provide information potentially relevant to predicting y_0.

We embed all $n+1$ models jointly via Eq. (2), fit a regressor $\hat{h}^{(t)} : \mathbb{R}^d \to \mathcal{Y}$ on the observed pairs $\{(\hat{\psi}_i, \tilde{y}_i^{(t)})\}_{i=1}^n$, and predict $\hat{y}_0^{(t)} = \hat{h}^{(t)}(\hat{\psi}_0)$. We note that Eq. (1) uses no score information, so the target model is embedded alongside the reference models. Out-of-sample extensions exist (see, e.g., Trosset and Priebe, 2008; Chen et al., 2022).

Helm et al. (2026) established that nearest neighbor regression using representations from paired DKPS is query efficient relative to the sample score. The main result of this section shows that benchmark prediction with PKPS is query-efficient relative to paired DKPS. Beyond Assumptions 1–4, our result requires a bounded and Lipschitz score on the population perspective space and a model distribution that has positive mass everywhere (Assumptions 5 and 6, Appendix D). The strict part of Definition 1 additionally requires that no two query designs share an atom, so that independently sampled query sets are disjoint almost surely (Assumption 7). In the following, MSE denotes mean squared error and m denotes a per-model query budget with $\min_i |Q_i| \geq m$. We assume the query kernel is from a family of bounded kernels indexed by a bandwidth σ whose $\sigma \to 0$ limit is such that $k_0(u, u') > 0$ if $u = u'$ and 0 otherwise (e.g., $k_\sigma(u, u') = \exp(-\|u - u'\|^2/2\sigma^2)$).

Definition 1 (Query efficiency; following Helm et al., 2026)**.** Fix a per-model query budget m and let h_n denote a score estimator using n reference models. The sequence $(h_1, h_2, \ldots)$ is *query-efficient relative to* $(h'_1, h'_2, \ldots)$ if there exists N such that $\mathrm{MSE}(h_n) \leq \mathrm{MSE}(h'_n)$ with high probability for all $n \geq N$, with the inequality strict for at least one $n^* \geq N$.

We write $h_{n,\sigma}^{(K)}$ for an estimator that uses a K-nearest-neighbor regressor on the PKPS constructed with a query kernel with bandwidth σ. Since k_0 only puts positive mass on identical queries, $h_{n,0}^{(K)}$ is K-nearest-neighbor regression on the paired DKPS. Let $\hat{h}_n^{(K)}$ denote the cross-validated PKPS estimator, where the bandwidth is selected on held-out reference models over a finite grid containing $\sigma = 0$. A nonzero bandwidth is chosen only when the estimated improvement exceeds the validation confidence radius (Appendix D).

Theorem 2 (PKPS is query efficient relative to paired DKPS)**.** *Suppose Assumptions 1–6 hold. For every pairing fraction $\rho \in [0, 1]$ and every K, with high probability for all n sufficiently large,* $\mathrm{MSE}\big(\hat{h}_n^{(K)}\big) \leq \mathrm{MSE}\big(h_{n,0}^{(K)}\big)$. *If Assumption 7 holds and $v := \mathrm{Var}(y_0) > 0$, then for every $0 < c < v$ there are budgets and a bandwidth $\sigma^\star$ such that, whenever the grid contains $\sigma^\star$, with high probability for all n sufficiently large,* $\mathrm{MSE}\big(\hat{h}_n^{(K)}\big) \leq \mathrm{MSE}\big(h_{n,0}^{(K)}\big) - c$. *In particular, $(\hat{h}_n^{(K)})$ is query-efficient relative to $(h_{n,0}^{(K)})$.*

Theorem 2 states that PKPS is never worse than paired DKPS at any pairing level. Further, under Assumption 7 it is strictly better than paired DKPS. Hence, PKPS is query-efficient relative to paired DKPS. We note that query-efficiency is transitive and that Theorem 2, together with Theorem 2 of Helm et al. (2026), implies that $\hat{h}_n^{(K)}$ is query-efficient relative to the sample score.

The same machinery covers (model, task) score completion ($m_i^{(t)} = 0$): since Eq. (1) uses only responses, a model that has answered none of task t's queries is embedded from its responses to other tasks, and the regression transfers the scores of reference models evaluated on t. We evaluate both query-efficiency and completion empirically in Section 4.

4 Experiments

We evaluate PKPS on synthetic data and on two large benchmark suites.

Synthetic data. We generate n models with latent ability vectors $v_i \in \mathbb{R}^5$ and T tasks with latent vectors $t_k \in \mathbb{R}^5$, scores $v_i \cdot t_k + \epsilon$ with additive noise ϵ, and queries drawn near each task center ($u_j = t_k + \eta_j$ with isotropic noise η_j). Each model answers m queries per task. We control unpairedness via $\rho = m_{ii'}/m$ and sparsity via a probability of observing a task p_{task} and, conditioned on observing a task, a probability of answering a query p_{query}.

Benchmark suites. We consider two large benchmark suites. The first consists of 18 tasks from HELM-Lite (Liang et al., 2023) – 7 math subjects (MATH, Hendrycks et al., 2021), 5 translation pairs (WMT-14, Bojar et al., 2014), medical multiple choice (MedQA, Jin et al., 2021), and 5 legal

subjects (LegalBench, Guha et al., 2023). We consider 93 models that have been evaluated on all 18 tasks and induce sparsity and unpairedness by subsampling each model's observed queries.

The second suite is built from the Every Eval Ever datastore (Batzner et al., 2026), a community-maintained collection of item-level evaluation results covering 36 benchmarks and 165 models (and growing). The store exhibits the sparsity and unpairedness that motivate this paper: only 15% of (model, benchmark) pairs have been evaluated and 59% of model pairs share no benchmark at all. Within a benchmark, pairedness is bimodal: on 26 of 33 tasks, models answered essentially identical item sets (mean $\rho_{ii'} > 0.99$), while the remaining seven – all arena-style (Chiang et al., 2024) – are severely unpaired (mean $\rho_{ii'}$ between 0.27 and 0.66). We study a dense 45 model and 16 task submatrix so that true scores are available and sparsity and pairedness can be controlled. The tasks include MATH-MC levels 1–5 (Hendrycks et al., 2021), GSM-MC (Cobbe et al., 2021), GPQA-Diamond (Rein et al., 2024), nine categories from JudgeBench (Tan et al., 2025) and RewardBench 2 (Malik et al., 2025).

Both suites were chosen because they include item-level responses – most large public benchmark leaderboards only include scores. Details describing the HELM and EEE data such as the included models, included tasks, the number of queries per task, etc. are provided in Appendices A and B, respectively.

Methods and baselines. We evaluate PKPS on both query-efficiency and (model, task) score completion. Unless otherwise stated we embed all responses and queries into a 3072-dimensional space via `gemini-embedding-001` (Lee et al., 2025), apply PCA separately to the collection of embedded queries and embedded responses, and use an RBF kernel on the reduced query space and a linear kernel on the reduced response space. Given the PKPS pairwise distance matrix, we embed the models in a $d = 8$ space and use K-nearest neighbor regression with $K = 5$.

Depending on the task we include different baselines. Each produces an estimate of $y_i^{(t)}$:

1. *Query-efficiency*. We compare PKPS to the sample score (i.e., $\tilde{y}_i^{(t)}$), paired DKPS (Helm et al., 2026), Item Response Theory ("IRT", Polo et al., 2024) and a convex combination of PKPS and sample score (i.e., $\alpha\, \tilde{y}_i^{(t)} + (1-\alpha)\, \hat{y}_{i,\mathrm{PKPS}}^{(t)}$).

2. *Score completion.* We compare PKPS to Low-Rank Matrix Completion ("LRMC", Zeng and Papailiopoulos, 2026) and a convex combination of LRMC and PKPS (i.e., $\alpha\, \hat{y}_{i,\mathrm{LRMC}}^{(t)} + (1-\alpha)\, \hat{y}_{i,\mathrm{PKPS}}^{(t)}$).

Both the query kernel RBF bandwidth and the convex coefficient for the ensemble are selected via cross-validation on the observed data. Following Zeng and Papailiopoulos (2026), all methods operate on a pre-processed score matrix: observed scores are logit-transformed, standardized per task, and centered by a two-way (model, task) additive bias. Following Helm et al. (2026), we use the leave-one-family-out protocol where all models from the target model's developer are excluded. As such, the reported errors are conservative. We report average mean absolute error (MAE) of the predicted score versus the true score for every setting over 16 seeds (50 on synthetic data). Additional experimental details are provided in Appendix A.

4.1 Results

PKPS is query-efficient on the synthetic data. Figure 2 shows the MAE for PKPS, DKPS, and the sample score for estimating a model's true score under the query-efficient evaluation protocol. Every observed cell of the score matrix is calculated with a budget of m queries. We vary the query budget (panel a), the pairwise query overlap (panel b), the number of models (panel c), the number of tasks (panel d), and the task coverage (panel e). Panel (f) studies score completion, where the targets are held-out cells and the score-only baseline is the task mean.

PKPS outperforms DKPS everywhere and the sample score at small query budgets (panel a). Panel (b) is the empirical counterpart of Theorem 2. At $\rho = 0$, DKPS is worse than the sample score, while a single shared query ($\rho = 0.25$ at $m = 4$) restores it to PKPS's level. PKPS is flat in ρ as it only requires only distributional overlap (Assumption 3). As n grows for a fixed sparsity and pairedness, PKPS can take advantage of additional reference models while DKPS cannot (panel c).

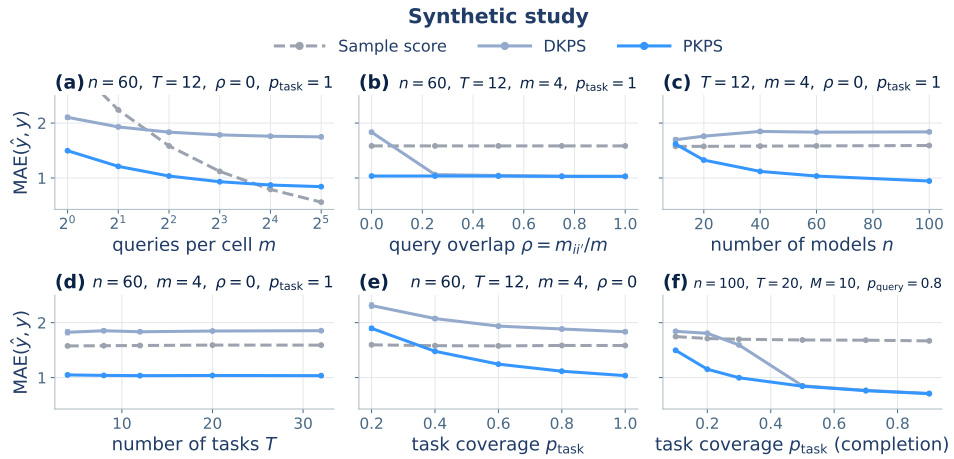

Figure 2: MAE on the synthetic benchmark (± 1 SEM). **(Panels a–e)** Query-efficient evaluation where every observed cell is measured with m queries. We vary the budget m (panel a), the cross-model overlap ρ (panel b), the cohort size n (panel c), the number of tasks T (panel d), and the task coverage p_{task} (panel e), at unpaired designs (ρ=0) except where ρ is swept. **(panel f)** Score completion of held-out cells as a function of task coverage.

Task coverage behaves similarly (panel e) and the number of tasks matters little for either method (panel d) — though this observation likely hinges on the ratio between task-level noise and query-level noise. In the completion panel (panel f), PKPS outperforms DKPS when observed cells are scarce and they are matched for moderate to large coverage.

PKPS-based methods dominate query-efficient evaluation. Figure 3a–c shows the MAE for PKPS, DKPS, IRT, the sample score, and the ensemble on the HELM suite. Each model answers its own independently sampled queries. At m=1, PKPS attains 0.137 and the ensemble 0.126 versus 0.292 for the sample score (Wilcoxon p=8×10^{-6} on average task error differences, better on all 18 tasks), so PKPS with one query matches the sample score at $m = 4$ (panel a). As the budget grows, the sample score's noise falls below the representations' residual bias and it eventually overtakes them. The ensemble follows the better of its two components throughout (panel a). PKPS improves with the number of reference models (b; $0.157 \rightarrow 0.124$) and with the task coverage (panel c; $0.161 \rightarrow 0.130$), while DKPS is flat. IRT is poorly identified without a shared item block and is worse than the sample score throughout.

The gains hold across settings. Figure 3d–g shows the distribution of the per-cell difference between the sample score's and the ensemble's absolute errors, by dataset (panel d), budget (panel e), cohort (panel f), and task coverage (panel g). The ensemble outperforms the sample score on 69% of cells (78% at m=1) and in the mean of every row. The gains are largest on the rich-response datasets (panel d) and at the smallest budgets (panel e), where a single response carries the most information, and they shrink toward zero as the budget grows. The mean gain is positive at every cohort size (panel f) and every coverage (panel g).

The same representations complete missing cells in the score matrix. Figure 4a–d shows the MAE for LRMC, PKPS, and the ensemble on missing cells when a fraction p_{task} of (model, task) cells is observed at query depth p_{query}, at cohort sizes n=93 (solid) and n=10 (dashed). No responses from a missing cell exist, so the sample score is unavailable. The ensemble matches or outperforms LRMC at every operating point, and the margin is largest when observations are scarce: at the small cohort it is more accurate at every task coverage (panel a), query depth (panel b), and number of tasks (panel c) (0.120 versus 0.139; Wilcoxon p=1×10^{-3}, better on 15 of 18 tasks). LRMC requires many models to estimate its factors: with n=10, additional tasks do not close the gap (panel c), and sweeping the number of reference models shows the ensemble is accurate from n=20 while LRMC approaches it only near the full cohort (panel d). At the full cohort the meth-

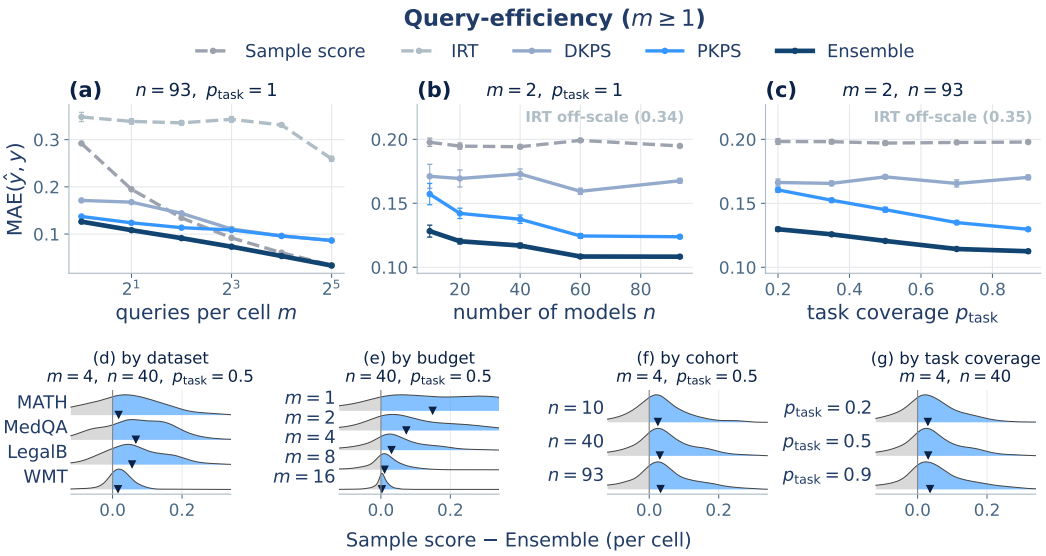

Figure 3: Query-efficiency on the HELM suite. **(a–c)** MAE against the true full-evaluation score when each model answers m unpaired queries per observed cell (± 1 SEM), as a function of budget m, cohort size n, and task coverage p_{task}. **(d–g)** Per-cell difference in MAE between Sample score and Ensemble (positive difference favors Ensemble) by dataset, budget, cohort, and coverage. ▼ is the mean for a given setting.

ods converge and the ensemble remains best on average (0.100 versus 0.105), although the per-task difference is not significant (better on 9 of 18 tasks).

The completion gains are modest but consistent. Figure 4e–h shows the distribution of the per-missing-cell difference between LRMC's and the ensemble's absolute errors, by dataset (panel e), cohort (panel f), coverage (panel g), and query depth (panel h). The win is smaller than in the query-efficiency task but the mean gain is positive in every row.

The results are qualitatively similar on the Every Eval Ever suite. The second suite shows the same behavior with larger margins (lever sweeps in Appendix B). At m=1, PKPS attains less than a third of the sample score's error (0.070 versus 0.224), below the accuracy the sample score reaches at m=8 (0.081). On missing cells, the ensemble outperforms LRMC everywhere, including the full cohort (Wilcoxon p=3 $\times$ 10^{-5}, better on every task), which is not significant on HELM. Table 1 shows the performance of all methods on both tasks and both suites. The ensemble is the best or tied in every column.

The results are insensitive to representation and pipeline choices. Varying the MDS dimension over 4–24 moves PKPS's MAE by less than 0.02, and the cross-validated bandwidth matches the best fixed choice (Appendix C). Ablations of the regression method, the response kernel, the ensemble weighting rule, and the incidental query overlap are also in Appendix C. Our observations are generally robust across these choices.

5 Discussion

In this paper we introduced the PKPS, a collection of low-dimensional representations of models defined on sparse and unpaired collections of embedded responses — a realistic setting outside of highly curated leaderboards and evaluation collections. We prove that the PKPS is query-efficient relative to its fully-paired counterpart and show that, as $\rho \to 0$, paired methods such as DKPS and IRT degrade sharply while PKPS does not. The same representations support both benchmark prediction tasks and the ensemble dominates both.

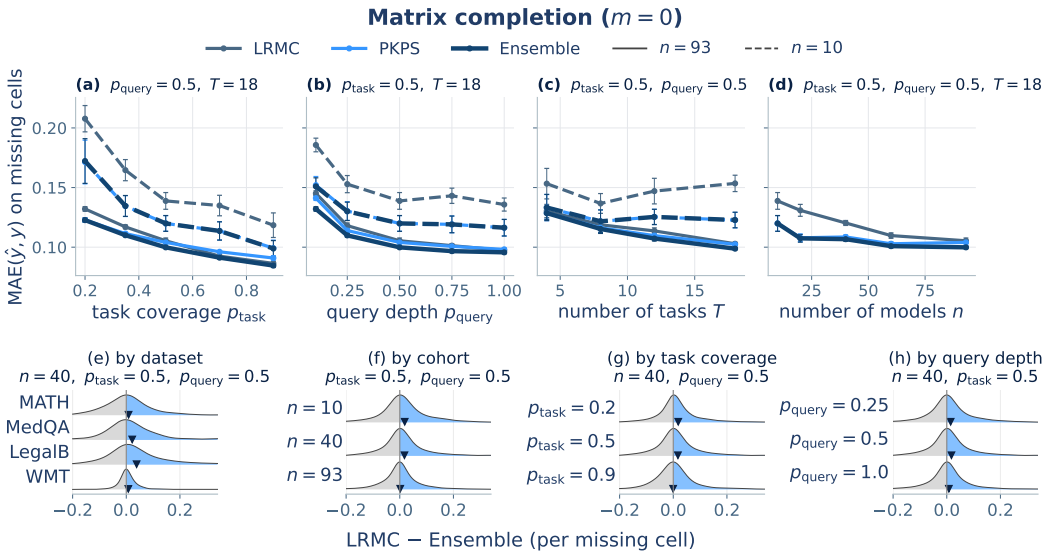

Figure 4: Score completion on the HELM suite. **(a–d)** MAE on missing cells (± 1 SEM) when a fraction p_{task} of (model, task) cells is observed at query depth p_{query}, as a function of task coverage, query depth, number of tasks, and cohort size. **(e–h)** Per-cell difference in MAE between LRMC and Ensemble (positive difference favors Ensemble) by dataset, cohort, coverage, and query depth. ▼ is the mean for a given setting.

Practical deployment. Our results suggest a simple workflow for practical deployment: (i) maintain the cached responses (and, optionally, per-response scores) that are evaluation produced when evaluating a model; (ii) fit PKPS on the cached responses; and (iii) predict with a simple regression in the perspective space, optionally ensembled with the sample score. As shared caches like the Every Eval Ever datastore grow, each new upload improves the perspective space, and the cumulative savings grow with them.

Limitations and future work. (i) *Stochastic scoring functions.* Our analysis assumes the mapping from a response to its score is deterministic. Many modern evaluation frameworks use LLM-as-judge scoring (Zheng et al., 2023), where scores are stochastic and may depend on contextual factors beyond the individual response. If each score takes the form $s(f(q)) + \epsilon$ with ϵ i.i.d., repeated scoring recovers the current setting in the limit and the extension is conceptually straightforward. (ii) *Beyond uniform observation.* Assumption 1 models observed cells as uniform subsamples, and our experiments induce sparsity and unpairedness by uniform masking. Missingness in deployed evaluation data is not uniform: which models are evaluated on which benchmarks correlates with developer, release date, and expected performance. Extending the analysis to modeled observation processes is a natural and important next step. (iii) *Unpaired monitoring in multi-agent systems.* Paired DKPS has been used as a tool for tracking the perspectives of multi-agent systems (Helm et al., 2024). These approaches require asking each agent a collection of fixed queries each time point and are prohibitively expensive at scale. The PKPS removes this requirement as agents can be monitored through the heterogeneous interactions they naturally produce.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods and simple data processing. We also used them in the creation of artifacts (e.g., using Matplotlib to create figures), and draft / edit parts of the manuscript with heavy user guidance. We have reviewed all AI-assisted work. For example, generated code was verified and tested for correctness by all authors. Generated text was heavily edited. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The main text and appendix provide enough information to reproduce all artifacts and results.

REFERENCES

Aranyak Acharyya, Michael W. Trosset, Carey E. Priebe, and Hayden S. Helm. Consistent estimation of generative model representations in the Data Kernel Perspective Space. *arXiv preprint arXiv:2409.17308*, 2024. doi: 10.48550/arXiv.2409.17308.

Aranyak Acharyya, Joshua Agterberg, Youngser Park, and Carey E. Priebe. Concentration bounds on response-based vector embeddings of black-box generative models. *arXiv preprint arXiv:2511.08307*, 2025. doi: 10.48550/arXiv.2511.08307.

Jan Batzner, Sree Harsha Nelaturu, Damian Stachura, et al. Every Eval Ever: A unifying schema and community repository for AI evaluation results, 2026. Dataset: https://huggingface.co/datasets/evaleval/EEE_datastore.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 2014. doi: 10.3115/v1/W14-3302.

Ingwer Borg and Patrick J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2nd edition, 2005. doi: 10.1007/0-387-28981-X.

Sam Bowyer, Acyr Locatelli, and Kris Cao. Efficient benchmarking is just feature selection and multiple regression. *arXiv preprint arXiv:2605.25773*, 2026. doi: 10.48550/arXiv.2605.25773.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023. doi: 10.48550/arXiv.2303.12712.

Guodong Chen, Hayden S. Helm, Kate Lytvynets, Weiwei Yang, and Carey E. Priebe. Mental state classification using multi-graph features. *Frontiers in Human Neuroscience*, 16, 2022. doi: 10.3389/fnhum.2022.930291.

Nan Chen, Hayden Helm, Youngser Park, Carey Priebe, and Soledad Villar. Extracting information from fine-tuned weights. In *NeurIPS Workshop on Non-Euclidean Foundation Models and Geometric Learning (NEGEL)*, 2025a.

Rick Chen, Joseph Ternasky, Afriyie Samuel Kwesi, Ben Griffin, Aaron Ontoyin Yin, Zakari Salifu, Kelvin Amoaba, Xianling Mu, Fuat Alican, and Yigit Ihlamur. VCBench: Benchmarking LLMs in venture capital. *arXiv preprint arXiv:2509.14448*, 2025b. doi: 10.48550/arXiv.2509.14448.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot Arena: An open platform for evaluating LLMs by human preference. In *International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arXiv.2403.04132.

François Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. ARC-AGI-2: A new challenge for frontier AI reasoning systems. *arXiv preprint arXiv:2505.11831*, 2025. doi: 10.48550/arXiv.2505.11831.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168.

Ivan Dokmanić, Reza Parhizkar, Juri Ranieri, and Martin Vetterli. Euclidean distance matrices: Essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015. doi: 10.1109/MSP.2015.2398954.

Brandon Duderstadt, Hayden S. Helm, and Carey E. Priebe. Comparing foundation models using data kernels. *arXiv preprint arXiv:2305.05126*, 2023. doi: 10.48550/arXiv.2305.05126.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Zambrano, et al. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.48550/arXiv.2308.11462.

Hayden Helm, Brandon Duderstadt, Youngser Park, and Carey E. Priebe. Tracking the perspectives of interacting language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. doi: 10.18653/v1/2024.emnlp-main.90.

Hayden Helm, Aranyak Acharyya, Youngser Park, Brandon Duderstadt, and Carey Priebe. Statistical inference on black-box generative models in the Data Kernel Perspective Space. In *Findings of the Association for Computational Linguistics (ACL)*, 2025. doi: 10.18653/v1/2025.findings-acl.204.

Hayden Helm, Ben Johnson, and Carey E. Priebe. Query-efficient model evaluation using cached responses. *arXiv preprint arXiv:2605.07096*, 2026. doi: 10.48550/arXiv.2605.07096.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks Track*, 2021. doi: 10.48550/arXiv.2103.03874.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

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. doi: 10.48550/arXiv.2509.11106.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The Platonic Representation Hypothesis. In *International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arXiv.2405.07987.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. FinanceBench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*, 2023. doi: 10.48550/arXiv.2311.11944.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14), 2021. doi: 10.3390/app11146421.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, et al. Gemini embedding: Generalizable embeddings from Gemini. *arXiv preprint arXiv:2503.07891*, 2025. doi: 10.48550/arXiv.2503.07891.

Yang Li, Jie Ma, Miguel Ballesteros, Yassine Benajiba, and Graham Horwood. Active evaluation acquisition for efficient LLM benchmarking. *arXiv preprint arXiv:2410.05952*, 2024. doi: 10.48550/arXiv.2410.05952.

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 (TMLR)*, 2023. doi: 10.48550/arXiv.2211.09110.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2308.03688.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. RewardBench 2: Advancing reward model evaluation. *arXiv preprint arXiv:2506.01937*, 2025. doi: 10.48550/arXiv.2506.01937.

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 *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024. doi: 10.18653/v1/2024.naacl-long.139.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with fewer examples. In *International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arXiv.2402.14992.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. doi: 10.48550/arXiv.2311.12022.

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. doi: 10.48550/arXiv.2405.10938.

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. doi: 10.48550/arXiv.2206.04615.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. JudgeBench: A benchmark for evaluating LLM-based judges. In *International Conference on Learning Representations*, 2025. doi: 10.48550/arXiv.2410.12784.

Warren S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4): 401–419, 1952. doi: 10.1007/BF02288916.

Michael W. Trosset and Carey E. Priebe. The out-of-sample problem for classical multidimensional scaling. *Computational Statistics & Data Analysis*, 52(10):4635–4642, 2008. doi: 10.1016/j.csda.2008.02.031.

Michael W. Trosset and Carey E. Priebe. Continuous multidimensional scaling. *arXiv preprint arXiv:2402.04436*, 2024. doi: 10.48550/arXiv.2402.04436.

Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Benjamin Recht. Low-rank solutions of linear matrix equations via Procrustes flow. In *International Conference on Machine Learning (ICML)*, 2016. doi: 10.48550/arXiv.1507.03566.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2024. doi: 10.18653/v1/2024.eacl-long.95.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. doi: 10.48550/arXiv.2206.07682.

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 (ICLR)*, 2025. doi: 10.48550/arXiv.2406.19314.

Qinyuan Ye, Harvey Yiyun 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. doi: 10.18653/v1/2023.findings-emnlp.503.

Yi Yu, Tengyao Wang, and Richard J. Samworth. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015. doi: 10.1093/biomet/asv008.

Yuchen Zeng and Dimitris Papailiopoulos. You don't need to run every eval. *arXiv preprint arXiv:2606.24020*, 2026. doi: 10.48550/arXiv.2606.24020.

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. doi: 10.48550/arXiv.2306.05685.

A Experimental details

Suite. 18 tasks over four datasets from HELM-Lite (Liang et al., 2023): MATH (7 subjects), WMT-14 (5 translation directions), MedQA, and LegalBench (5 subsets), restricted to the 93 models evaluated on all four. Table 2 lists every task with its query count; the loader subsamples each task to at most 120 queries with a fixed seed, so datasets contribute comparably.

Table 2: HELM suite composition: 18 tasks over four datasets, with the number of unique queries per task. Every task is scored for all 93 shared models; the loader caps each task at 120 queries (seeded subsample).

Dataset	Task	Queries	Dataset	Task	Queries
MATH	algebra	135	WMT-14	cs–en	873
MATH	counting & probability	39	WMT-14	de–en	776
MATH	geometry	38	WMT-14	fr–en	754
MATH	intermediate algebra	52	WMT-14	hi–en	387
MATH	number theory	30	WMT-14	ru–en	613
MATH	prealgebra	86	MedQA	med_qa	1000
MATH	precalculus	57	LegalBench	abercrombie	95
LegalBench	corporate lobbying	490	LegalBench	citizenship questions	1000
LegalBench	function of decision section	367	LegalBench	proa	95

Model coverage (HELM). The 93 shared models span 20 developers: `01-ai_yi-34b`, `01-ai_yi-6b`, `01-ai_yi-large-preview`, `AlephAlpha_luminous-base`, `AlephAlpha_luminous-extended`, `AlephAlpha_luminous-supreme`, `ai21_j2-grande`, `ai21_j2-jumbo`, `ai21_jamba-1.5-large`, `ai21_jamba-1.5-mini`, `ai21_jamba-instruct`, `allenai_olmo-7b`, `amazon_nova-lite-v1`, `amazon_nova-micro-v1`, `amazon_nova-pro-v1`, `anthropic_claude-2.0`, `anthropic_claude-2.1`, `anthropic_claude-3-5-haiku-20241022`, `anthropic_claude-3-5-sonnet-20240620`, `anthropic_claude-3-5-sonnet-20241022`, `anthropic_claude-3-haiku-20240307`, `anthropic_claude-3-opus-20240229`, `anthropic_claude-3-sonnet-20240229`, `anthropic_claude-instant-1.2`, `anthropic_claude-instant-v1`, `anthropic_claude-v1.3`, `cohere_command`, `cohere_command-light`,

cohere_command-r, cohere_command-r-plus, databricks_dbrx-instruct, deepseek-ai_deepseek-llm-67b-chat, deepseek-ai_deepseek-v3, google_gemini-1.0-pro-001, google_gemini-1.0-pro-002, google_gemini-1.5-flash-001, google_gemini-1.5-flash-002, google_gemini-1.5-pro-001, google_gemini-1.5-pro-002, google_gemini-1.5-pro-preview-0409, google_gemini-2.0-flash-exp, google_gemma-2-27b-it, google_gemma-2-9b-it, google_gemma-7b, google_text-bison@001, google_text-unicorn@001, meta_llama-2-13b, meta_llama-2-70b, meta_llama-2-7b, meta_llama-3-70b, meta_llama-3-8b, meta_llama-3.1-405b-instruct-turbo, meta_llama-3.1-70b-instruct-turbo, meta_llama-3.1-8b-instruct-turbo, meta_llama-3.2-11b-vision-instruct-turbo, meta_llama-3.2-90b-vision-instruct-turbo, meta_llama-3.3-70b-instruct-turbo, meta_llama-65b, microsoft_phi-2, microsoft_phi-3-medium-4k-instruct, microsoft_phi-3-small-8k-instruct, mistralai_mistral-7b-instruct-v0.3, mistralai_mistral-7b-v0.1, mistralai_mistral-large-2407, mistralai_mistral-medium-2312, mistralai_mixtral-8x22b, mistralai_mixtral-8x7b-32kseqlen, mistralai_open-mistral-nemo-2407, nvidia_nemotron-4-340b-instruct, openai_gpt-3.5-turbo-0613, openai_gpt-4-0613, openai_gpt-4-1106-preview, openai_gpt-4-turbo-2024-04-09, openai_gpt-4o-2024-05-13, openai_gpt-4o-2024-08-06, openai_gpt-4o-mini-2024-07-18, openai_text-davinci-002, openai_text-davinci-003, qwen_qwen1.5-110b-chat, qwen_qwen1.5-14b, qwen_qwen1.5-32b, qwen_qwen1.5-72b, qwen_qwen1.5-7b, qwen_qwen2-72b-instruct, qwen_qwen2.5-72b-instruct-turbo, qwen_qwen2.5-7b-instruct-turbo, snowflake_snowflake-arctic-instruct, tiiuae_falcon-40b, tiiuae_falcon-7b, upstage_solar-pro-241126, writer_palmyra-x-004, writer_palmyra-x-v2, writer_palmyra-x-v3.

Representations. All queries and responses are embedded with `gemini-embedding-001` (3072-d). MATH and WMT-14 responses are free text and are embedded directly. MedQA and LegalBench responses are single-token answers (multiple-choice letters and class indices); each is standardized to the text it denotes before embedding—the option text the letter selects, recovered from the MedQA source data and verified against model behavior on 999 of 1000 instances, and the class name the index encodes. Queries share one PCA space (capped at 64 components) and responses another, both fit on observed rows only; the response space's dimension is set to the total that separate per-dataset reductions would retain (second Zhu–Ghodsi elbow per dataset, at most 48 components each; 92 on this suite), and response vectors are unit-normalized. Nothing separates the datasets by construction: the query kernel, whose bandwidth is set from within-dataset distances, determines how much weight cross-dataset comparisons receive.

Kernels. The query kernel is an RBF whose bandwidth is selected per run by leave-one-model-out K-NN (K=8) on the observed scores, over the grid $\sigma \in \{0.03, 0.1, 0.3, 1, 3\} \times \text{median}$ pairwise query distance plus the near-delta (DKPS) limit; Appendix C shows the cross-validated bandwidth tracks the best fixed choice. The response kernel is linear on real data and RBF on synthetic. Each (model, query) pair uses one sampled response (r=1).

Model embedding. Models are embedded by classical MDS with d=8 throughout, following Helm et al. (2026) and fixed a priori (d = latent dimension on synthetic); Appendix C probes the choice of d.

Regressor. All embedding methods share one regressor pipeline: logit-transformed scores, per-task standardization to unit variance, a two-way (model + task) additive bias, and a distance-weighted K-NN (K=5) regression of the residual on the model embedding, fit *leave-one-family-out*: all models from the target's developer are excluded from its training set, so predictions never lean on near-duplicate siblings (Helm et al., 2026). Held-out cells are predicted directly; observed cells leave their own target out. The pipeline is identical to the LRMC baseline's, so the two methods differ only in how the residual is modeled (K-NN on the perspective versus a low-rank factorization) and performance differences are attributable to the representation. Matching the per-task standardization is important: without it, the model offset is dominated by the tasks with the largest logit variance.

Baselines. Low-rank matrix completion (LRMC) follows BenchPress (Zeng and Papailiopoulos, 2026): logit space, per-task standardization, two-way bias, rank-2 ALS with ridge λ=0.1, averaged over 5 random initializations; on observed cells it is cross-fit so a cell's prediction never sees its own sample. IRT is a 1PL (Rasch) model on binary tasks: item difficulties by maximum likelihood (L-BFGS) on the largest fully-observed (model $\times$ item) block, per-model ability by bounded scalar MLE, clipped to $[-4, 4]$. The sample baseline is the cell's own m-query mean.

Ensemble. On observed cells, the prediction is $\alpha\,\hat{y}_{\text{sample}} + (1-\alpha)\,\hat{y}_{PKPS}$ with a scalar α per held-out family, selected on a 21-point grid by leave-one-family-out validation against the other families' cached full scores (the same information the regressor trains on); on missing cells, PKPS and LRMC are blended with a weight fit by held-out cross-validation against observed scores. No quantity ever uses the scores of the family being predicted.

Sweeps and reporting. All real-data sweeps use 16 seeds; curves and table entries report the per-seed mean over tasks, then the mean ($\pm$1 SEM) over seeds. Query efficiency: budgets $m \in \{1, 2, 4, 8, 16, 32\}$ at full cohort and coverage; cohort $n \in \{10, 20, 40, 60, 93\}$ (HELM) or $\{10, 20, 30, 45\}$ (EEE) and task coverage $p_{\text{task}} \in \{0.2, 0.35, 0.5, 0.7, 0.9\}$, both at the few-query point m=2. Completion: task coverage as above and query depth $p_{\text{query}} \in \{0.1, 0.25, 0.5, 0.75, 1.0\}$ with cohort lines $n \in \{10, \text{full}\}$; task count $T \in \{4, 8, 12, 18\}$ (HELM) or $\{4, 8, 12, 16\}$ (EEE); the cohort sweep holds p_{task}=p_{query}=0.5.

Statistical tests. Paired differences between methods are assessed with two-sided Wilcoxon signed-rank tests on per-task error differences: for each task we average the per-cell absolute-error difference over models and seeds, and test the per-task means against zero (18 tasks on HELM, 16 on EEE). The query-efficiency tests compare the ensemble to the sample score at m=1 (full cohort and coverage); the completion tests compare the ensemble to LRMC at the central operating point (p_{task}=p_{query}=0.5) at each cohort size. The per-cell improvement fractions quoted in Section 4.1 are computed from the same per-cell records, pooled over the conditional-figure sweeps.

Synthetic study. 50 seeds per configuration; latent dimension 5, observed embedding dimension 20, response noise 0.5; both kernels RBF. Query-efficiency panels (Figure 2a–e): every observed cell carries m queries and a sample score with measurement noise $4/\sqrt{m}$; the true score is predicted by a distance-weighted average of the K=8 nearest models' sample scores, restricted to cells they observe. Completion panel (f): score noise 0.1, held-out cells predicted by K-NN regression with K=5. Unswept levers default to n=60, T=12, m=4, ρ=0, p_{task}=1; panel-specific settings are given in each figure title.

Compute. All experiments run on a single multi-core CPU node; text embeddings are precomputed once through the embedding API and cached. Embedding the EEE suite (30,304 pooled responses plus 7,665 queries, $\sim$11M tokens) took $\sim$13 minutes and under $2 of API credit; each 16-seed sweep completes in minutes (EEE) to hours (HELM) on the same node.

B The Every Eval Ever suite

The second suite is built from the Every Eval Ever datastore (Batzner et al., 2026), an MIT-licensed, community-maintained collection of item-level evaluation results in a unified format (inputs, raw model outputs, and per-item scores). The full store covers 36 benchmarks and 165 models with only 15% of (model, benchmark) pairs evaluated (median 5 benchmarks per model), and 59% of model pairs share no benchmark. Within a benchmark, 26 of the 33 with at least two models are essentially fully paired (23 with identical item sets across every model pair), while the seven arena-style benchmarks (six Fibble arenas and Wordle arena) are natively unpaired: mean per-pair overlap 0.27–0.66, median zero on six of the seven, and per-model coverage of the item union 0.15–0.33—in interactive evaluation, the queries a model sees depend on its matchups. We take the five benchmarks with the widest item-level model coverage—MATH-MC, GSM-MC, GPQA-Diamond, JudgeBench, and RewardBench 2—restricted to the 45 models present on all five, for which every model answered every item; sub-tasks (MATH-MC levels 1–5; four JudgeBench and five RewardBench 2 categories) give 16 tasks (Table 3). Every domain is free text (including the judging benchmarks' rationales), so all

responses and queries are embedded with `gemini-embedding-001` directly, and the response space is built exactly as in Appendix A: one shared PCA over all five benchmarks (dimension 83, matched to the per-benchmark total), unit-normalized, with the query kernel gating cross-benchmark comparisons. Each cell's observable pool is 32 queries drawn with a model-dependent seed, so two models share only incidental queries (median $\rho_{ii'}^{(t)} = 0.09$, mean 0.18, largest on the smallest tasks); the query-efficiency budget m and the completion depth p_{query} subsample this pool, and the prediction target is always the cell's full-depth mean over all items, which is never embedded. All scores are binary, so IRT applies to every task (on the HELM suite it excludes WMT). Figures 5 and 6 repeat the lever sweeps of Figures 3 and 4 on this suite; Table 1 summarizes.

Table 3: EEE suite composition: 16 tasks over five benchmarks, with the number of items per (model, task) cell. Every task is fully answered by all 45 shared models; each cell's observable pool is 32 items, sampled per model.

Benchmark	Task	Items	Benchmark	Task	Items
MATH-MC	level 1	430	JudgeBench	coding	42
MATH-MC	level 2	882	JudgeBench	knowledge	154
MATH-MC	level 3	1115	JudgeBench	math	56
MATH-MC	level 4	1199	JudgeBench	reasoning	98
MATH-MC	level 5	1288	RewardBench 2	factuality	475
GSM-MC	gsm_mc	1319	RewardBench 2	focus	495
GPQA-Diamond	gpqa_diamond	198	RewardBench 2	math	183
RewardBench 2	precise if	160	RewardBench 2	safety	450

Model coverage (EEE). The 45 models shared by all five benchmarks span 13 developers:
`cohere/c4ai-command-a-03-2025`, `cohere/c4ai-command-r-08-2024`, `cohere/c4ai-command-r-plus-08-2024`, `cohere/c4ai-command-r7b-12-2024`, `cohere/command-a-reasoning-08-2025`, `deepseek/deepseek-r1-0528`, `deepseek/deepseek-v3-1-terminus`, `deepseek/deepseek-v3-2`, `deepseek/deepseek-v3-2-speciale`, `deepseek/deepseek-v4-flash-fp8`, `deepseek/deepseek-v4-pro`, `google/gemini-3-1-pro-preview`, `google/gemma-2-27b-it`, `google/gemma-2-9b-it`, `google/gemma-3-12b-it`, `google/gemma-3-27b-it`, `google/gemma-4-e2b-it`, `google/gemma-4-e4b-it`, `llm360/k2-v2-instruct`, `meta/llama-4-maverick-17b-128e-instruct-fp8`, `meta/meta-llama-3-1-8b-instruct`, `minimax/minimax-m2-5`, `mistral/mistral-small-4-119b-2603`, `moonshot/kimi-k2-5`, `moonshot/kimi-k2-6`, `moonshot/kimi-k2-thinking`, `openai/gpt-5-5`, `openai/gpt-5-mini`, `openai/gpt-5-nano`, `openai/gpt-oss-120b`, `openai/gpt-oss-20b`, `qwen/qwen3-30b-a3b-thinking-2507`, `qwen/qwen3-5-122b-a10b`, `qwen/qwen3-5-2b`, `qwen/qwen3-5-397b-a17b`, `qwen/qwen3-6-35b-a3b`, `qwen/qwen3-6-plus`, `qwen/qwen3-next-80b-a3b-thinking`, `stepfun/step-3-5-flash`, `xiaomi/mimo-v2-flash`, `zai-org/glm-4-6-fp8`, `zai-org/glm-4-7-flash`, `zai-org/glm-4-7-fp8`, `zai-org/glm-5-1-fp8`, `zai-org/glm-5-fp8`.

C Sensitivity analyses

Figure 7 probes the two representation hyperparameters on the query-efficiency experiment (16 seeds throughout).

MDS dimension (a, b). Repeating the budget sweep at $d \in \{4, 8, 24\}$, the d=4 and d=8 curves are within 0.003 of each other at every budget on both suites (at the smallest budgets, smaller d is very slightly better—fewer coordinates to estimate from one noisy query per cell), while d=24 costs up to 0.013 at the smallest EEE budgets and up to 0.018 at the largest HELM budgets; the paper's choice (d=8, following Helm et al., 2026) is never the worst point at any budget.

Query bandwidth (c). Fixing the bandwidth at each grid multiplier instead of cross-validating traces the expected curve: near-delta multipliers and the exact-matching limit ($k_Q(u, u') = \mathbf{1}\{u = u'\}$, abbreviated δ) are substantially worse (unpaired pools leave them little to compare), broad multipliers are best, and the cross-validated choice matches the best fixed multiplier on both suites

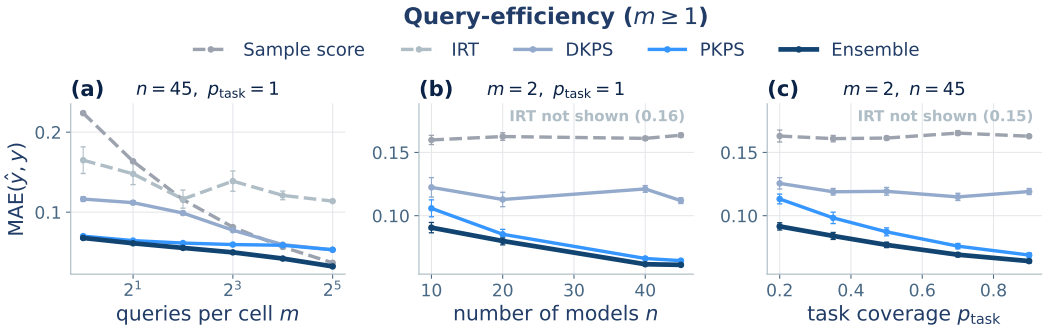

Figure 5: Query-efficient evaluation on the Every Eval Ever suite (45 models, 16 tasks; MAE against the true full-evaluation score, ± 1 SEM over 16 seeds). The protocol matches Figure 3. The Ensemble is the most accurate at every budget and PKPS at every budget up to m=16; DKPS trails because the query pools are unpaired, and IRT is poorly identified.

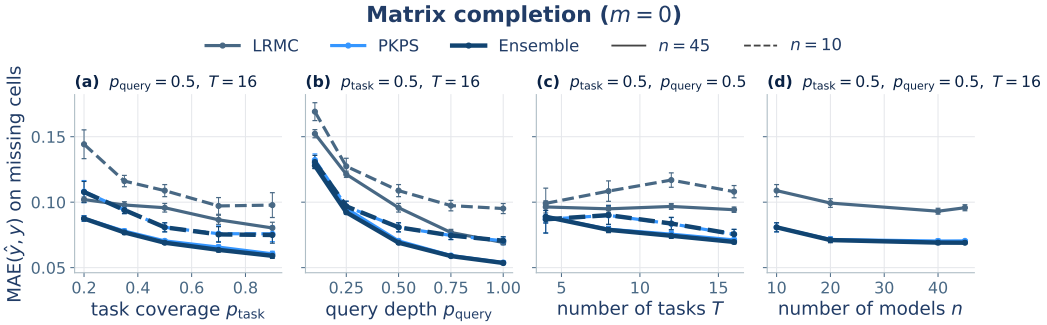

Figure 6: Score completion on the Every Eval Ever suite (MAE on missing cells, ± 1 SEM; solid n=45, dashed n=10). The protocol matches Figure 4. Unlike on the HELM suite, PKPS and the Ensemble outperform LRMC at every operating point.

(0.124 vs. 0.123 on HELM, 0.065 vs. 0.064 on EEE)—the one hyperparameter that matters is the one the method selects automatically. Sensitivity to the number of reference models is part of the main experiments (the cohort sweeps of Figures 3b and 4d).

Pipeline ablations. Repeating the budget sweep with one choice changed at a time: the cross-validated ensemble weight is best or tied at every budget on both suites (the inverse-variance alternative loses 0.007 on HELM and 0.012 on EEE at m=1); replacing the K-NN regressor with least squares helps slightly on HELM (0.132 vs. 0.137 at m=1) but hurts on EEE (0.061 vs. 0.053 at m=32); an RBF response kernel changes PKPS by at most 0.002; and enforcing exactly disjoint EEE query pools (true $\rho = 0$) changes PKPS by at most 0.006 at $m \leq 8$.

D Proofs of Theorems 1 and 2

Formal definitions and assumptions. We state the formal statements referenced in Section 3.

Definition 2 (Population PKPS)**.** Let each model i's (task, query, response) triples be drawn i.i.d. from a design distribution P_i: each triple independently draws a task, then a query conditioned on the task, then the model's response. Write $z = (u, x_i(u))$ for an embedded query–response pair (response noise included). The *population affinity* is

$$A^*_{ii'} = \frac{\mathbb{E}\left[k_Q(u,u')\, k_R(x_i(u), x_{i'}(u'))\right]}{\mathbb{E}\left[k_Q(u,u')\right]}, \qquad z \sim P_i,\ z' \sim P_{i'} \text{ independent},$$

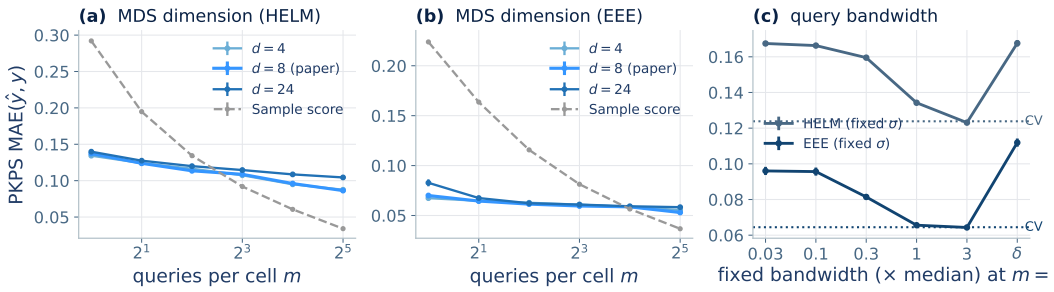

Figure 7: Sensitivity analyses on the query-efficiency experiment (±1 SEM, 16 seeds). **(a, b)** PKPS MAE against budget at MDS dimension $d \in \{4, 8, 24\}$ (the paper's choice d=8 drawn heavier; sample dashed); d=4 and d=8 are nearly indistinguishable on both suites, while d=24 trails at the extremes. **(c)** MAE at m=2 with the query bandwidth fixed at each grid multiplier (δ = exact matching); dotted lines mark the cross-validated selection, which matches the best fixed choice on both suites.

the population squared distances are $D^{*2}(i,i') = A^*_{ii} + A^*_{i'i'} - 2A^*_{ii'}$, and the *population perspectives* $\psi_1, \ldots, \psi_n \in \mathbb{R}^d$ (the rows of $\Psi^* \in \mathbb{R}^{n \times d}$) are the rank-$d$ spectral embedding of the centered Gram matrix $B^* = -\frac{1}{2} J D^{*2} J$, where $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$.

Assumption 1 (Observation model)**.** Each scoring function is the mean of bounded per-query scores, and each observed cell $Q_i^{(t)}$ is a uniform subsample of $Q^{(t)}$ at the budget m.

Assumption 2 (Boundedness)**.** Kernels and embedded responses are bounded.

Assumption 3 (Population overlap)**.** The *cross-model query-kernel mass* is bounded below: $\kappa = \inf_{i,i'} \mathbb{E}[k_Q(u,u')] > 0$.

Assumption 4 (Spectral identifiability)**.** B^* has rank d with spectral gap $\lambda_d > 0$.

Assumption 5 (Score regularity)**.** The benchmark score is bounded and γ-Lipschitz on the population perspective space.

Assumption 6 (Model support)**.** The model distribution places mass on every neighborhood of the population perspective space.

Assumption 7 (Pairing deficiency)**.** The cross-model mass of the exact-matching kernel vanishes: $\kappa_0 = \sup_{i \neq i'} \mathbb{E}[k_0(u,u')] = 0$ for independent queries $u \sim P_i$, $u' \sim P_{i'}$, where $k_0(u,u') = \mathbf{1}\{u = u'\}$; equivalently, no two query designs share an atom.

Population quantities. Throughout, the quantities of Definition 2 are used as stated there. By Assumption 2 the kernels are bounded, $k_Q \in [0,1]$ after normalization (e.g., RBF) and $|k_R| \leq B_R$, and by Assumption 4 the rank-d spectral embedding exists with $\lambda_d(B^*) > 0$.

Lemma 1 (Affinity concentration)**.** *Under Assumptions 2 and 3, for models i, i' with $\min(|Q_i|, |Q_{i'}|) \geq m$ and any $\delta \in (0,1)$ we have, with probability at least $1-\delta$,*

$$|A_{ii'} - A^*_{ii'}| \;\leq\; \frac{2(1+B_R)}{\kappa^2}\, t_m(\delta), \qquad t_m(\delta) = \bar{B}\sqrt{\frac{\log(4/\delta)}{2m}}, \quad \bar{B} = \max(2B_R, 1),$$

provided $t_m(\delta) \leq \kappa/2$.

Proof. The numerator $N = \frac{1}{|Q_i||Q_{i'}|}\sum_{j,l} k_Q(u_j, u'_l)\, k_R(x_{ij}, x_{i'l})$ and denominator $Z = \frac{1}{|Q_i||Q_{i'}|}\sum_{j,l} k_Q(u_j, u'_l)$ of Eq. (1) are two-sample U-statistics of order $(1,1)$ in the i.i.d. pairs $z_j \sim P_i$, $z'_l \sim P_{i'}$, with kernels of range at most $2B_R$ and 1 respectively (the response draw is part of z, so no per-query response replication is needed). By Hoeffding's inequality for two-sample U-statistics (Hoeffding, 1963, Section 5), each deviates from its mean by more than its range times $\sqrt{\log(4/\delta)/2m}$ with probability at most $\delta/2$; since $\bar{B}$ dominates both ranges, a union bound gives $|N - N^*| \leq t_m(\delta)$ and $|Z - Z^*| \leq t_m(\delta)$ jointly with probability $\geq 1-\delta$. On this event, since $Z^* \geq \kappa$ by Assumption 3 and $t_m(\delta) \leq \kappa/2$, we have $Z \geq \kappa/2$, and

$$|A_{ii'} - A^*_{ii'}| = \frac{|NZ^* - N^*Z|}{ZZ^*} \;\leq\; \frac{|N - N^*|}{Z} + \frac{|N^*|\,|Z - Z^*|}{ZZ^*} \;\leq\; \frac{2\,t_m(\delta)}{\kappa} + \frac{2B_R\, t_m(\delta)}{\kappa^2} \;\leq\; \frac{2(1+B_R)}{\kappa^2}\, t_m(\delta),$$

using $|N^*| \le B_R$ and $\kappa \le 1$.

□

Lemma 2 (Perspective perturbation). *Let* $\varepsilon_\infty = \max_{i,i'} |A_{ii'} - A^*_{ii'}|$. *There is an orthogonal* $W \in \mathbb{R}^{d\times d}$ *such that*

$$\max_i \|\hat{\psi}_i - W\psi_i\| \;\le\; \frac{25\sqrt{d}\, n\, \varepsilon_\infty}{\sqrt{\lambda_d(B^*)}}.$$

Proof. Entrywise, $|D^2 - D^{*2}| \le 4\varepsilon_\infty$, and double centering inflates an entrywise bound by at most a factor of 4 (each entry of JMJ is a signed combination of an entry, a row mean, a column mean, and the grand mean of M), so $E = B - B^*$ satisfies $\|E\|_{\mathrm{op}} \le \|E\|_F \le 8\, n\, \varepsilon_\infty$. Let B_d be the top-d truncation of B that classical MDS embeds ($B_d = \hat{\Psi}\hat{\Psi}^\top$, positive semidefinite). By Weyl's inequality every eigenvalue of B beyond the d-th, and every negative eigenvalue, has magnitude at most $\|E\|_{\mathrm{op}}$ (since $\lambda_{d+1}(B^*) = 0$ and $B^* \succeq 0$), so $\|B_d - B\|_{\mathrm{op}} \le \|E\|_{\mathrm{op}}$. The matrix $B_d - B^*$ has rank at most $2d$, hence

$$\|B_d - B^*\|_F \;\le\; \sqrt{2d}\,\|B_d - B^*\|_{\mathrm{op}} \;\le\; \sqrt{2d}\,\big(\|B_d - B\|_{\mathrm{op}} + \|E\|_{\mathrm{op}}\big) \;\le\; 2\sqrt{2d}\,\|E\|_F.$$

Both $B_d = \hat{\Psi}\hat{\Psi}^\top$ and $B^* = \Psi^*\Psi^{*\top}$ are positive semidefinite of rank at most d, so by the Procrustes alignment lemma of Tu et al. (2016, Lemma 5.4),

$$\min_{W\in O(d)} \|\hat{\Psi} - \Psi^* W\|_F^2 \;\le\; \frac{\|B_d - B^*\|_F^2}{2(\sqrt{2}-1)\,\lambda_d(B^*)}.$$

Combining, $\max_i \|\hat{\psi}_i - W\psi_i\| \le \|\hat{\Psi} - \Psi^* W\|_F \le \frac{2\sqrt{2d}}{\sqrt{2(\sqrt{2}-1)}} \cdot \frac{8n\varepsilon_\infty}{\sqrt{\lambda_d(B^*)}} \le \frac{25\sqrt{d}\, n\varepsilon_\infty}{\sqrt{\lambda_d(B^*)}}$. □

Proof of Theorem 1. Apply Lemma 1 at confidence δ/n^2 to each of the at most n^2 model pairs; a union bound gives, with probability at least $1-\delta$,

$$\varepsilon_\infty \;=\; \max_{i,i'} |A_{ii'} - A^*_{ii'}| \;\le\; \frac{2(1+B_R)\bar{B}}{\kappa^2}\sqrt{\frac{\log(4n^2/\delta)}{2m}}.$$

Plugging this into Lemma 2 yields $\max_i \|\hat{\psi}_i - W\psi_i\| \le C\kappa^{-2}\lambda_d^{-1/2}\sqrt{d}\, n\sqrt{\log(4n^2/\delta)/2m}$ with $C = 50(1+B_R)\bar{B}$, which depends only on the kernel bounds. Lemma 1 requires $t_m(\delta/n^2) \le \kappa/2$; in the complementary case the claimed bound already exceeds the trivial one—every perspective satisfies $\|\psi_i\|^2 \le \lambda_1(B) \le 8B_R n$ and likewise for $\hat{\psi}_i$, so $\max_i \|\hat{\psi}_i - W\psi_i\| \le 6\sqrt{B_R n}$, while the right-hand side is at least $C\kappa^{-1}\sqrt{dn}/(2\bar{B}\sqrt{8B_R}) \ge 6\sqrt{B_R n}$ once C also dominates $96B_R\bar{B}$—so the theorem holds for every m after enlarging C by a constant factor. □

The remainder of this appendix proves Theorem 2. The first claim is a cross-validation transfer argument with a guarded selection rule. The second claim combines a lower bound on the MSE of every paired estimator under Assumption 7 with the achievability lemmas below (Lemmas 3 and 4).

Proof of Theorem 2, first claim. Split the reference models into a training set T and a validation set V with $|V| = n_v$. For each σ in the finite grid $\Sigma \ni 0$, build the perspective space on T, embed any model outside T out of sample by its Eq. (1) comparison row to T, and let $\hat{y}_\sigma$ be K-nearest-neighbor regression on T's (perspective, score) pairs, with validation loss $\hat{L}(\sigma) = \frac{1}{n_v}\sum_{i\in V}(\hat{y}_\sigma(f_i) - y_i)^2$. Let $\tau = \sqrt{\log(2|\Sigma|/\delta')/2n_v}$ be the validation confidence radius. The guarded selection is

$$\hat{\sigma} = \begin{cases} \arg\min_{\sigma\in\Sigma} \hat{L}(\sigma) & \text{if } \min_\sigma \hat{L}(\sigma) \le \hat{L}(0) - 2\tau, \\ 0 & \text{otherwise.} \end{cases}$$

Conditional on T, the validation losses are i.i.d. across the draw of validation models and bounded in $[0,1]$, so by Hoeffding's inequality and a union bound over Σ, with probability at least $1-\delta'$, $\sup_{\sigma\in\Sigma} |\hat{L}(\sigma) - L(\sigma)| \le \tau$, where $L(\sigma)$ is the MSE on a fresh target. On this event, if $\hat{\sigma} \ne 0$ then

$$L(\hat{\sigma}) \;\le\; \hat{L}(\hat{\sigma}) + \tau \;\le\; \hat{L}(0) - 2\tau + \tau \;\le\; L(0),$$

and if $\hat{\sigma} = 0$ the two estimators coincide; either way $\mathrm{MSE}(\hat{h}_n^{(K)}) \le \mathrm{MSE}(h_{n,0}^{(K)})$, since the $\sigma = 0$ member coincides with paired DKPS on the shared queries with the same regressor (Section 3.2). This holds at every (m, ρ). □

Lemma 3 (Achievability with exact reference scores)**.** *Suppose the reference scores are observed exactly ($\tilde{y}_i^{(t)} = y_i$). Under Assumptions 1–6, for any $\epsilon > 0$ there exist a bandwidth σ and (n, m), with m polynomial in $(1/\epsilon, 1/\kappa, n)$ and independent of ρ, at which $\mathrm{MSE}(h_{n,\sigma}^{(1)}) \le \epsilon$ with high probability.*

Proof of Lemma 3. Take scores in $[0, 1]$ without loss of generality (Assumption 5). Fix $\epsilon > 0$ and set targets $c \le \sqrt{\epsilon/2}/(8\gamma)$ for the estimation error and $\epsilon' \le \sqrt{\epsilon/2}/(2\gamma)$ for the population nearest-neighbor distance. By Assumption 6 there is $n_0(\epsilon')$ such that for $n \ge n_0$, a reference model within population distance ϵ' of the target exists except with probability at most $\epsilon/8$. Fix such an n. Applying Lemma 1 with confidence $\delta = \epsilon/(8n^2)$ to every pair, a union bound and Lemma 2 give $\max_i \|\hat{\psi}_i - W\psi_i\| \le c$ except with probability at most $\epsilon/8$, once

$$m \;\ge\; C' \,\frac{d\,(1 + B_R)^2\,\bar{B}^2\,n^2 \log(n/\epsilon)}{\kappa^4\,\lambda_d\,c^2} \;=\; \mathrm{poly}(n, 1/\kappa, 1/\epsilon),$$

for an absolute constant C' (this choice also satisfies the proviso $t_m \le \kappa/2$ of Lemma 1), and does not involve the pairing fraction ρ. On the intersection of the two events, for the target model f_0 with nearest estimated reference neighbor f^* (perspectives $\psi^*, \hat{\psi}^*$; the target's $\psi_0, \hat{\psi}_0$),

$$|\hat{y}(f_0) - y_0| \;\le\; \gamma\,\|\psi^* - \psi_0\| \;\le\; \gamma\big(\|\hat{\psi}^* - W\psi^*\| + \|\hat{\psi}^* - \hat{\psi}_0\| + \|\hat{\psi}_0 - W\psi_0\|\big) \;\le\; \gamma\,(2c + \hat{\delta}),$$

using Assumption 5 and orthogonal invariance. The estimated nearest-neighbor distance $\hat{\delta}$ is at most the estimated distance to the ϵ'-close reference, itself at most $\epsilon' + 2c$ by the same alignment argument, so $|\hat{y} - y| \le \gamma(4c + \epsilon') \le \sqrt{\epsilon/2}$. Squaring, and adding the two failure events (squared errors are bounded by 1), $\mathrm{MSE}(\hat{y}) \le \epsilon/2 + \epsilon/4 \le \epsilon$. □

Lemma 4 (Achievability with noisy reference scores)**.** *Under Assumptions 1–6, for any $\epsilon > 0$ there exist a bandwidth σ and (n, m, K_n), with m polynomial in $(1/\epsilon, 1/\kappa, n)$, $K_n \to \infty$, and $K_n/n \to 0$, at which $\mathrm{MSE}(h_{n,\sigma}^{(K_n)}) \le \epsilon + \sigma_q^2/(mK_n)$ with high probability.*

Proof. Write the scoring function as $s^{(t)}(f, A) = \frac{1}{|A|}\sum_{q \in A} r(f, q)$ with per-query scores $r \in [0, 1]$, and let each reference cell $Q_i^{(t)}$ be a uniform subsample of $Q^{(t)}$ of size m (Assumption 1). Then $\mathbb{E}[\tilde{y}_i^{(t)} \mid f_i] = y_i$ and $\mathrm{Var}(\tilde{y}_i^{(t)} \mid f_i) \le \sigma_q^2/m$, and the noises $\varepsilon_i = \tilde{y}_i^{(t)} - y_i$ are independent across i because the subsamples are drawn independently (unpaired designs). For the K-nearest-neighbor regressor, decompose $\hat{y}(f_0) - y_0 = \frac{1}{K}\sum_{j \in N_K}(y_j - y_0) + \frac{1}{K}\sum_{j \in N_K}\varepsilon_j$ over the neighbor set N_K. The first term is bounded as in the proof of Lemma 3: perspective estimation error (Theorem 1) plus the population K-th-neighbor distance, which Assumption 6 makes small for n large with $K/n \to 0$, times the Lipschitz constant. The second term is a mean of K independent, bounded, mean-zero variables, so its conditional variance is at most $\sigma_q^2/(mK)$ and it concentrates by Hoeffding. Choosing $K_n \to \infty$ with $K_n/n \to 0$ (e.g., $K_n = \lceil \log n \rceil$) gives $\mathrm{MSE}(h_{n,\sigma}^{(K_n)}) \le \epsilon + \sigma_q^2/(mK_n) \to \epsilon$. □

Remark 1. *Under a common (paired) subsample the ε_i share the draw of the query set, and $\mathrm{Cov}(\varepsilon_i, \varepsilon_j)$ equals the variance of the common query-difficulty component, which the neighbor average does not reduce. The independence of the reference noises is where unpairedness* helps.

Contrast with paired DKPS. Under the limit kernel k_0 ($k_0(u, u') > 0$ iff $u = u'$), only identical query pairs enter the double sum, so the U-statistic runs over the ρm shared queries and the rate in Lemma 1 becomes $O_P\big((\rho m)^{-1/2}\big)$: DKPS's guarantee requires $\rho m \to \infty$ and vacates under Assumption 7, where the cross-model mass κ_0 is zero and its distance is undefined (Section 3).

Proof of Theorem 2, second claim. Floor. Under Assumption 7, $\Pr(q_j = q'_l) = 0$ for every pair of independently drawn queries, so almost surely every identity-matched sum is empty: any measurable

completion of a $\sigma = 0$ estimator predicts as a function of the reference data alone, independent of the target. Hence $\mathrm{MSE}(h_{n,0}^{(K)}) = v + \mathbb{E}[(\mathbb{E}y_0 - \hat{y})^2] \geq v$ (indeed for every neighbor count and every completion). *Achievability.* Fix $0 < c < v$ and set $\epsilon_0 = (v - c)/2$. Apply Lemma 4 at the fixed K with growing budgets: the perspective and neighbor-distance terms are handled exactly as there (Assumption 3 supplies $\kappa > 0$), and the noise term $\sigma_q^2/(m_n K)$ vanishes as $m_n \to \infty$. This yields a bandwidth $\sigma^\star$ and budgets m_n polynomial in $(n, 1/\kappa, 1/\epsilon_0)$ such that $L(\sigma^\star) \leq \epsilon_0/2$ with high probability for n large. Include $\sigma^\star$ in the grid, and choose n_v large enough that $3\tau \leq v - \epsilon_0/2$ and $2\tau \leq \epsilon_0/2$. On the deviation event of the first claim's proof, the guard triggers: $\hat{L}(\sigma^\star) \leq L(\sigma^\star) + \tau \leq \epsilon_0/2 + \tau \leq \hat{L}(0) - 2\tau$, using $\hat{L}(0) \geq L(0) - \tau \geq v - \tau$. Then $\mathrm{MSE}(\hat{h}_n^{(K)}) \leq \hat{L}(\hat{\sigma}) + \tau \leq \epsilon_0/2 + 2\tau \leq \epsilon_0 \leq v - c \leq \mathrm{MSE}(h_{n,0}^{(K)}) - c$. The margin $c > 0$ supplies the strict clause of Definition 1. □