BenchPress: Predicting LLM Benchmark Scores via Low-Rank Matrix Completion

Technical Report | February 2026

Abstract

Large language models are evaluated on an ever-growing set of benchmarks, yet no single model is tested on all of them. The resulting data forms a sparse matrix with roughly two-thirds of entries missing. We construct a fully-cited benchmark matrix spanning 83 frontier LLMs across 49 benchmarks (1,375 observed scores, 33.8% fill rate) and develop a matrix completion framework to predict the missing 2,692 cells. Through a systematic search over 34 candidate methods, we find that a logit-space blend of benchmark regression and rank-2 SVD -- BenchPress -- achieves 7.25% median absolute percentage error (MedAPE) on held-out scores under per-model leave-50%-out evaluation (3 seeds). The logit transform alone accounts for 70% of the gain by linearizing ceiling and floor effects on percentage-scale benchmarks. Analysis of the matrix's singular value spectrum on a 31x10 complete submatrix reveals strong low-rank structure: the first principal component explains 71.3% of the variance, and the first three capture 90.6%. A set of just five benchmarks can predict the remaining 44 with ~7.8% MedAPE. We additionally show that Claude Sonnet 4.5, when given the full matrix as context, achieves competitive MedAPE -- surpassing all statistical methods on random holdout -- at a cost of $0.86 per evaluation run.

1. Introduction

The evaluation landscape for large language models (LLMs) has expanded dramatically. As of early 2026, frontier models are routinely assessed on dozens of benchmarks spanning mathematics, coding, scientific reasoning, agentic task completion, multimodal understanding, and instruction following. However, no single model is evaluated on every benchmark. Model developers typically report results on a curated subset -- often chosen to highlight strengths -- while independent leaderboards cover different subsets. The result is a sparse matrix: models form the rows, benchmarks form the columns, and roughly two-thirds of the entries are missing.

This sparsity creates practical problems. Comparing models requires restricting to shared benchmarks, discarding valuable information. Practitioners choosing between models for deployment must navigate incomplete scorecards. And the research community lacks a unified picture of how capabilities relate across the evaluation landscape.

Matrix completion -- predicting missing entries from observed ones -- offers a natural solution. If benchmark scores exhibit low-rank structure (i.e., a small number of latent factors explain most of the variation), then the missing entries can be recovered with reasonable accuracy. This paper investigates whether this is the case, and if so, which methods perform best.

Contributions.

We make five main contributions:

1. A fully-cited benchmark matrix. We construct an 83 x 49 matrix of LLM benchmark scores, where every entry carries a citation URL to its original source. The matrix covers 21 model providers and 11 benchmark categories, with 1,375 observed entries (33.8% fill rate).

2. Logit-space matrix completion. We introduce the logit transform as a preprocessing step for benchmark score prediction. Working in log-odds space for percentage-scale benchmarks improves both regression and SVD methods by 11-13% relative, and implicitly handles bimodal benchmark distributions.

3. A systematic 34-method search. We evaluate 34 prediction methods across three rounds of exploration, testing six hypotheses about "wasted information" (model metadata, benchmark categories, non-linear transforms, bimodal handling, gradient-boosted trees, and missingness patterns). The winner, BenchPress, combines logit-space benchmark regression (alpha=0.6) with logit-space rank-2 SVD (alpha=0.4).

4. LLMs as benchmark predictors. We test whether LLMs themselves can predict benchmark scores by providing the full matrix as context. Claude Sonnet 4.5 achieves competitive MedAPE, surpassing all statistical methods on random holdout, at a cost of $0.86.

5. Structural findings. The matrix exhibits strong low-rank structure: the first principal component of a complete submatrix explains 71.3% of the variance. Five benchmarks suffice to predict the remaining 44 with ~7.8% MedAPE under proper holdout evaluation.

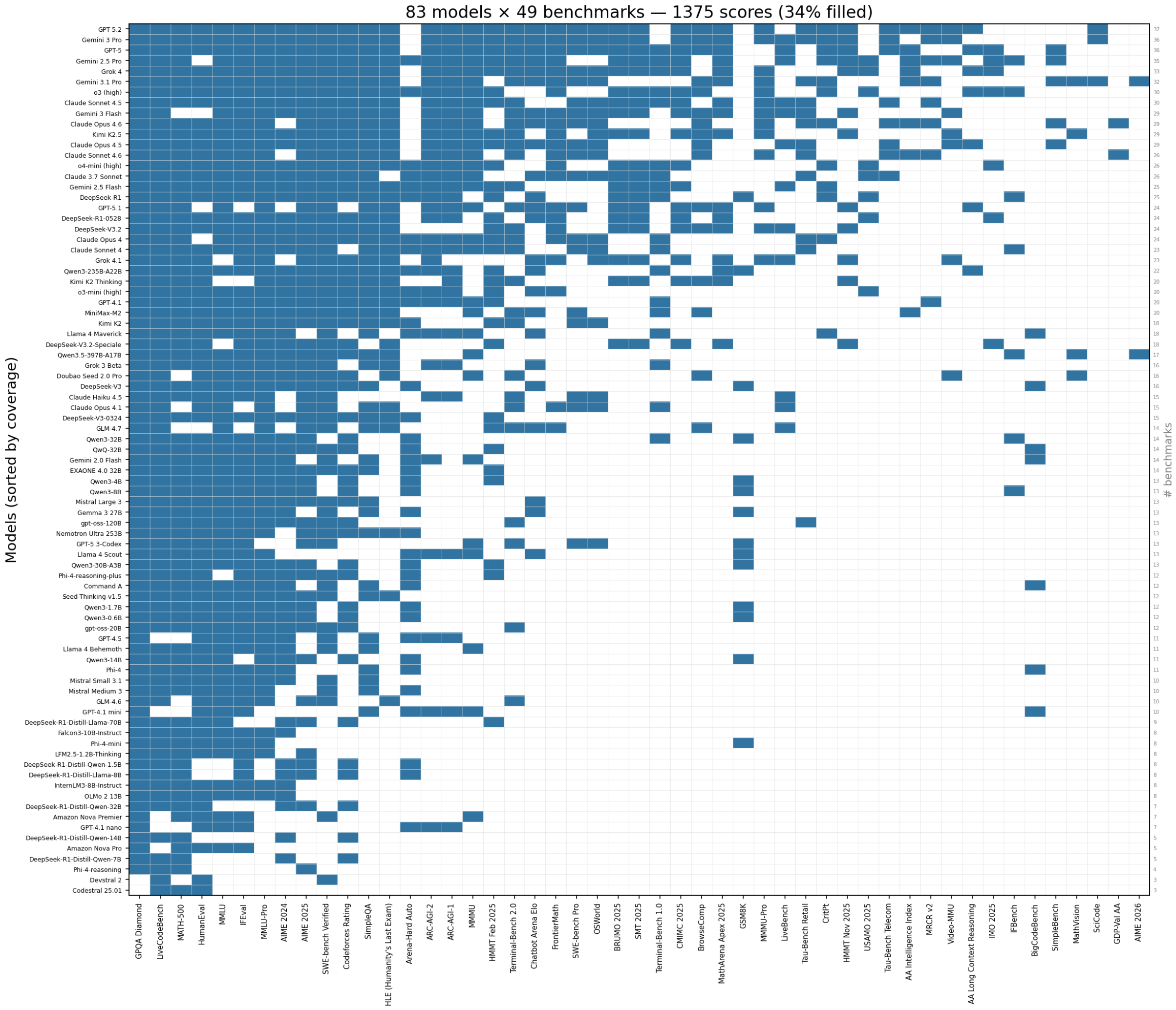

Figure 1: Sparsity pattern of the benchmark matrix. Blue cells indicate observed scores; white cells are missing. The matrix is 33.8% filled.

2. Dataset Construction

2.1 Model Selection

The matrix includes 83 instruct- or chat-tuned LLMs from 21 providers, released between January 2025 and February 2026. The models span a wide range of scales (0.6B to ~2T mixture-of-experts parameters), training paradigms (dense, MoE, distillation), and capability modes (57 reasoning models, 26 non-reasoning; 46 open-weight, 37 proprietary).

Provider	Count	Notable Models
OpenAI	13	o3, o4-mini, GPT-4.1, GPT-4.5, GPT-5
DeepSeek	12	R1, V3, R1-Distill family (1.5B-70B)
Alibaba (Qwen)	10	Qwen3 family (0.6B-235B), QwQ-32B
Anthropic	9	Claude Opus 4.6, Sonnet 4.5, Haiku 3.5
Google	7	Gemini 2.5 Pro/Flash, Gemini 3.1 Pro
Mistral	5	Mistral Large 3, Small 3.1
Microsoft	4	Phi-4, Phi-4-mini
Meta	3	Llama 4 Maverick/Scout, Llama 3.3 70B
xAI	3	Grok 3, Grok 3 mini
Others	17	Moonshot, ByteDance, Amazon, etc.

2.2 Benchmark Selection

We include 49 benchmarks spanning 11 categories. Most benchmarks report accuracy on a 0-100% scale. Exceptions include Chatbot Arena (Elo, ~1000-1400), Codeforces (rating, ~800-2200), and FrontierMath (0-25% accuracy range). Five benchmarks exhibit bimodal score distributions -- ARC-AGI-1, ARC-AGI-2, IMO 2025, USAMO 2025, and MathArena Apex 2025 -- where models either score near zero or well above 10%.

Category	Count	Examples
Math	15	AIME 2024/2025, MATH-500, FrontierMath, HMMT
Coding	7	LiveCodeBench, SWE-bench Verified, HumanEval
Agentic	6	SWE-bench Pro, Tau-Bench, Terminal-Bench
Reasoning	4	GPQA Diamond, ARC-AGI-1/2, HLE
Knowledge	4	MMLU, MMLU-Pro, SimpleQA, BrowseComp
Multimodal	3	MMMU, MMMU-Pro, Video-MMU
Instruction	3	IFEval, IFBench, Arena-Hard Auto
Science	2	GPQA Diamond, CritPt
Long Context	2	MRCR v2, LongBench v2
Composite	2	Chatbot Arena Elo, MathArena Apex
Human Pref.	1	Chatbot Arena Elo

2.3 Sparsity Structure

The matrix is 33.8% filled (1,375 / 4,067 cells). Coverage is uneven: the most-evaluated benchmarks (GPQA Diamond: 81 models, MMLU: 78, LiveCodeBench: 75) are an order of magnitude more covered than the sparsest (BRUMO: 5, Terminal-Bench 2.0: 6, SMT 2025: 7). Similarly, frontier models tend to be evaluated more broadly

(Gemini 2.5 Pro: 36 benchmarks) than smaller or niche models (some have only 4-6). This non-random missingness creates a "rich get richer" pattern that affects method choice: regression-based methods require correlated benchmark scores to exist in the training rows.

3. Evaluation Protocol

3.1 Per-Model Leave-50%-Out (Primary)

For each model with >=8 known scores, we randomly hide 50% of its observed scores, train the prediction method on the remaining scores plus all other models' complete data, and predict the hidden cells. We repeat this procedure across 3 random seeds and report the global median absolute percentage error (MedAPE) over all held-out cells pooled across seeds.

This protocol tests the realistic scenario: given some benchmark results for a model, predict the rest. It is more demanding than random holdout because it requires generalization within a single model's profile.

3.2 Random 20% Holdout (Secondary)

We randomly hide 20% of all 1,375 observed cells, predict them, and report MedAPE. This tests general matrix completion ability without the per-model constraint.

3.3 Extended Metrics

Beyond MedAPE, we report six additional metrics to capture different aspects of prediction quality:

- MAE: Mean absolute error in raw score points.

- +/-3 pts: Fraction of predictions within 3 score points of the true value.

- +/-5 pts: Fraction within 5 points.

- Bimodal Accuracy: Classification accuracy on five bimodal benchmarks (ARC-AGI-1/2, IMO 2025, USAMO 2025, MathArena Apex), thresholding at 10%.

- Coverage: Fraction of test cells receiving a finite prediction.

Why MedAPE?

We choose median (not mean) APE as the primary metric because benchmark score distributions are heavy-tailed: a few outlier benchmarks (e.g., Elo-scale, bimodal) can dominate mean-based metrics. MedAPE is robust to these outliers while remaining interpretable as "for a typical held-out score, the prediction is off by X%."

4. Methods

4.1 Baselines

Benchmark Mean (B0).

Predict each missing cell as the mean of observed scores for that benchmark. This ignores model identity entirely and serves as a floor.

k-Nearest Neighbors (B2).

For a missing entry (i, j), find the k=5 models most similar to model i (cosine similarity on shared benchmarks) and average their scores on benchmark j.

Benchmark-KNN (B3).

Transpose the KNN idea: find the 5 benchmarks most similar to j and predict model i's score as the average of its scores on those benchmarks.

4.2 Matrix Factorization

SVD (Soft-Impute).

We use the iterative soft-impute algorithm: initialize missing values with column means, compute a rank-r SVD, replace missing values with the rank-r reconstruction, and iterate until convergence. We test ranks r in {2, 3, 5, 8, 10}.

4.3 Regression-Based Methods

BenchReg.

For each target benchmark j, identify the k=5 most correlated benchmarks (among those with sufficient shared observations). Fit a Ridge regression from these 5 predictors to target j using all models with complete data. Predict missing values.

4.4 The Logit Transform

The single most impactful methodological choice is applying the logit transform to percentage-scale benchmark scores before regression or factorization:

```
logit(p) = log(p / (1-p)),  p in (0, 1)
```

where p = score/100, clipped to [0.005, 0.995] to avoid infinities.

Why it works.

Consider a benchmark like MMLU where frontier models score 88-92%. In raw space, the relationship between model capability and score is compressed near the ceiling: a model that is "twice as capable" might only gain 2 percentage points. In logit space, this ceiling effect is linearized. The logit transform correctly models that improving from 88% to 92% represents more "difficulty" than 48% to 52%.

Bimodal handling.

For benchmarks with bimodal distributions (e.g., ARC-AGI-2 where models score either ~0% or >10%), the logit transform maps the two modes far apart (logit(0.005) = -5.3 vs. logit(0.25) = -1.1), allowing SVD and regression to naturally separate them.

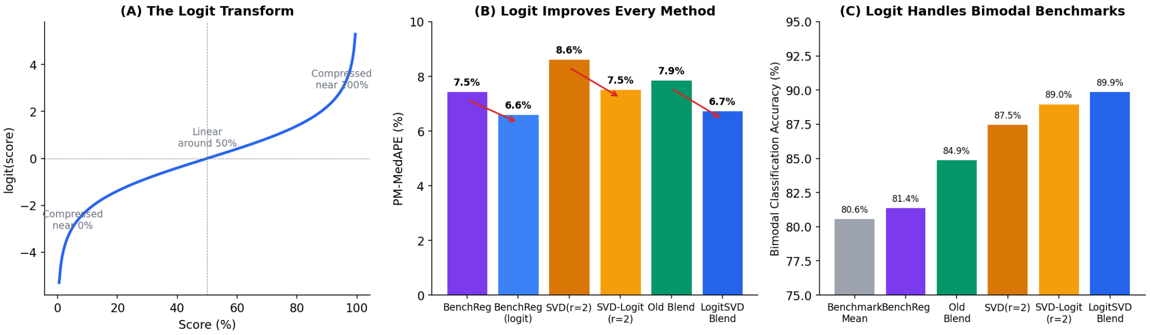

Figure 2: Effect of the logit transform. Left: the logit function mapping [0,100] to (-inf, +inf). Center: relative improvement from the logit transform. Right: bimodal classification accuracy improvement.

4.5 BenchPress: The Best Method

Our best method, BenchPress (formerly LogitSVD Blend), is a weighted average of two complementary predictors:

```
y_hat = 0.6 * y_LogitBR + 0.4 * y_SVD-Logit
```

The algorithm proceeds in five steps:

1. Identify percentage benchmarks. Any benchmark where all observed values fall in [-1, 101] is treated as percentage-scale (~92% of benchmarks).

2. LogitBenchReg. For each percentage benchmark j: transform observed scores to logit space, find the top-5 most correlated benchmarks (in logit space), fit Ridge regression, predict missing values, and apply the inverse logit (sigmoid) to return to [0, 100]. For non-percentage benchmarks, standard z-score BenchReg is used.

3. SVD-Logit(r=2). Transform all percentage columns to logit space, z-score normalize, run soft-impute SVD with rank 2, and inverse-transform back.

4. Blend. Where both predictors produce estimates, take the weighted average. Where only SVD-Logit exists (~21.5% of cells), use SVD-Logit alone. Where neither exists (~0.3%), fall back to the column mean.

5. Clamp. Clip all predictions to valid ranges: [0, 100] for percentage benchmarks.

Why this blend?

BenchReg exploits local correlations ("AIME 2024 predicts AIME 2025"), while SVD exploits global low-rank structure ("this model's overall profile resembles GPT-4.1"). These are complementary error profiles. The prior KNN-based blend partner made correlated errors with BenchReg because both are local similarity methods.

5. Results

5.1 Main Results (Post-Audit)

Table 1 presents the post-audit evaluation for BenchPress under per-model leave-50%-out evaluation (3 seeds).

Metric	BenchPress (Post-Audit)
MedAPE	7.25%
MAE	4.71
Within +/-3 pts	36.5%

Within +/-5 pts	51.7%
Bimodal Accuracy	94.2%
Coverage	99.7%
N (held-out cells)	1,944
Seeds	3

Table 1: Post-audit per-model leave-50%-out results. BenchPress achieves 7.25% MedAPE with 94.2% bimodal accuracy and 99.7% coverage across 1,944 held-out cells.

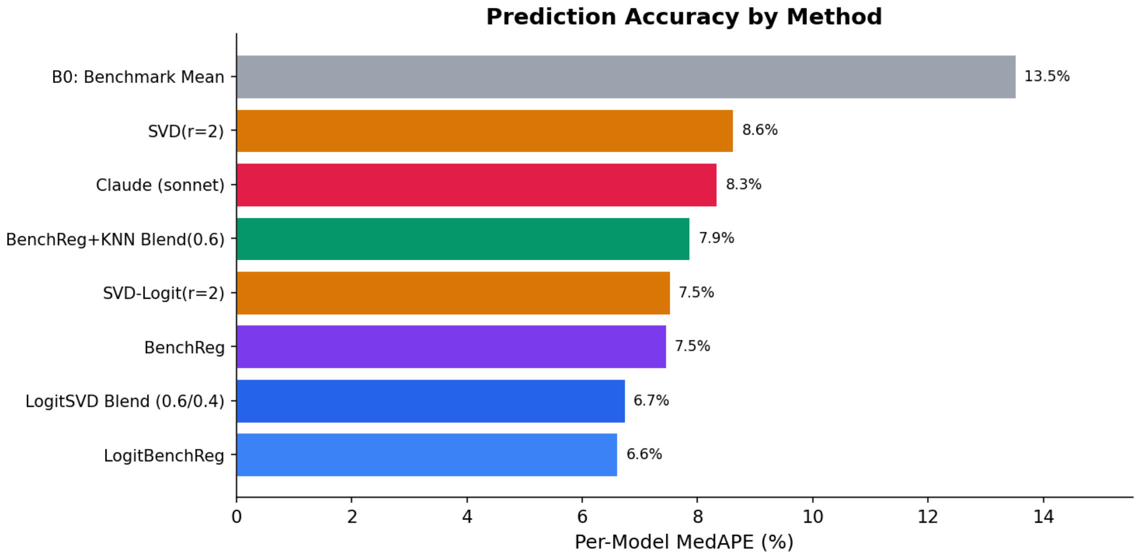

Figure 3: Method comparison. Per-model MedAPE for all evaluated methods. Lower is better.

5.2 Full 19-Method Baseline Comparison

Table 2 shows the complete ranking of all 19 baseline methods evaluated with the original single-seed evaluation harness.

Rank	Method	PM-MedAPE	R-MedAPE	R^2
1	BenchReg(k=5)	7.69%	5.86%	0.976
2	LogBenchReg	7.72%	6.27%	0.972
3	BenchReg+KNN	7.88%	6.43%	0.979
4	LogBlend	7.97%	6.32%	0.977
5	SVD(r=2)	8.09%	7.32%	0.979
6	Ensemble(avg3)	8.30%	7.09%	0.974
7	SVD(r=3)	9.07%	8.02%	0.972
8	Bench-KNN(k=5)	9.10%	8.07%	0.974
9	KNN(k=5)	9.32%	7.63%	0.970
10	Quantile+SVD5	9.64%	7.31%	0.957
11	NucNorm	9.70%	8.44%	0.953
12	Model-Normalized	10.00%	9.61%	0.962
13	PMF(r=5)	10.29%	8.48%	0.960

14	NMF(r=5)	10.46%	8.84%	0.951
15	SVD(r=5)	10.95%	8.79%	0.947
16	LogSVD(r=5)	12.42%	8.75%	0.935
17	Benchmark Mean	12.89%	11.71%	0.928
18	SVD(r=8)	13.18%	9.71%	0.914
19	SVD(r=10)	13.40%	10.33%	0.923

5.3 Decomposing the Improvement

The improvement from BenchReg+KNN Blend to BenchPress can be decomposed into two independent changes:

Change	From	To	Relative Delta
Add logit to BenchReg	7.45%	6.61%	-11.3%
Add logit to SVD	8.62%	7.52%	-12.8%
Replace KNN with SVD-Logit	7.86%	BenchPress	-14.2%

The logit transform consistently improves both regression and factorization methods by 11-13% relative. The additional gain from replacing KNN with SVD-Logit comes from decorrelating the blend components.

5.4 What Didn't Work

Model metadata (provider, parameters, reasoning mode), gradient-boosted trees, stacking, missingness features, and explicit bimodal handling all added noise or overfit on the sparse matrix. The consistent pattern is that additional features and complexity hurt.

Approach	MedAPE	Failure Mode
MetaKNN	8.57%	Metadata adds noise
MultiRidge	9.85%	Too many features for few rows
MissingnessKNN	9.25%	NaN pattern not informative
GBT per-benchmark	8.50%	Overfits with <40 training rows
BimodalAware	7.91%	Logit handles this better
MetaLearnerV2	7.14%	Stacking overfits

6. Intrinsic Dimensionality and Latent Structure

6.1 Singular Value Spectrum

The SVD spectrum of the 31 x 10 mean-centered complete submatrix reveals strong low-rank structure. The first principal component alone explains 71.3% of the variance.

Components	Cumulative Variance Explained
1	71.3%
1-2	83.0%
1-3	90.6%

Table 3: SVD spectrum of the 31x10 complete submatrix. The first three components capture over 90% of the variance.

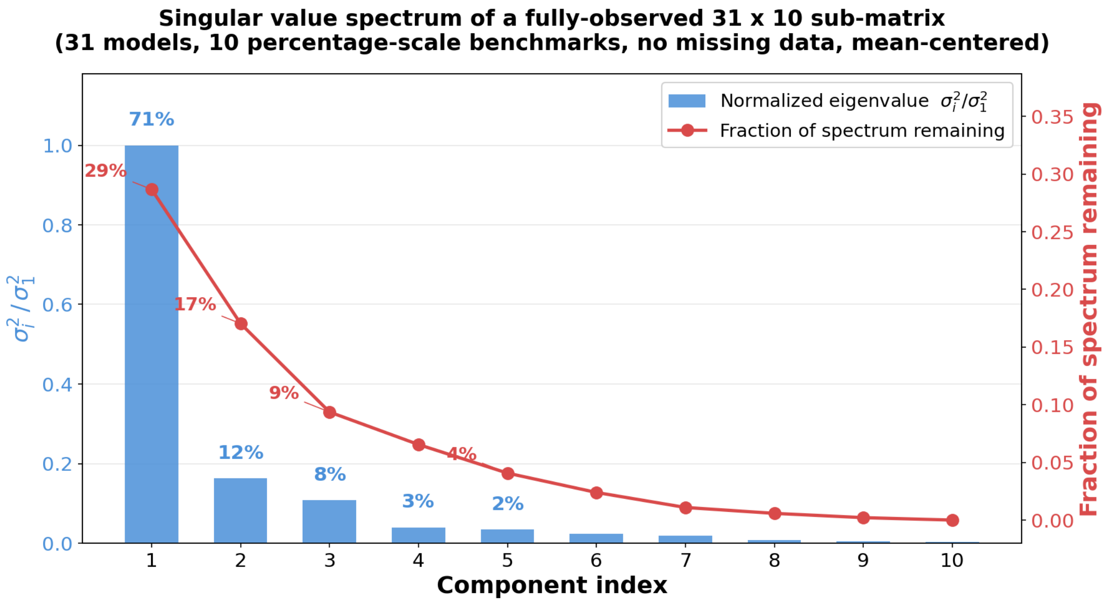

Figure: Singular value spectrum of the largest fully-observed sub-matrix (31 models x 10 benchmarks), mean-centered.
Blue bars: each squared singular value normalized by the largest. Red curve: fraction of total spectrum not yet captured by the top i components.
The first component captures 71% of the spectrum; two components capture 83%. By component 5, only 4% remains.
This steep decay — from raw data with no imputation — confirms the approximate low-rank structure of LLM benchmark scores.

Figure 4: SVD spectrum of the complete submatrix showing strong low-rank structure.

6.2 Factor Interpretation

Factor 1 -- "General Capability".

Top loadings: GPQA Diamond, LiveCodeBench, MMLU-Pro, MMLU, MATH-500. This is the dominant axis of LLM capability. The strongest models (o3, Claude Opus 4.6, GPT-5, Gemini 2.5 Pro) score uniformly high across all benchmarks.

Factor 2 -- "Frontier Reasoning".

Positive loadings: SimpleQA, ARC-AGI-2, HLE, FrontierMath, SWE-bench Verified. This factor distinguishes models that excel on genuinely novel, hard tasks from those that perform well on established benchmarks.

Practical meaning.

That the matrix has strong low-rank structure means: knowing just a few numbers about a model allows prediction of its performance across all 49 benchmarks with ~7% median error. The remaining variance consists of benchmark-specific noise and idiosyncratic model behaviors that do not generalize.

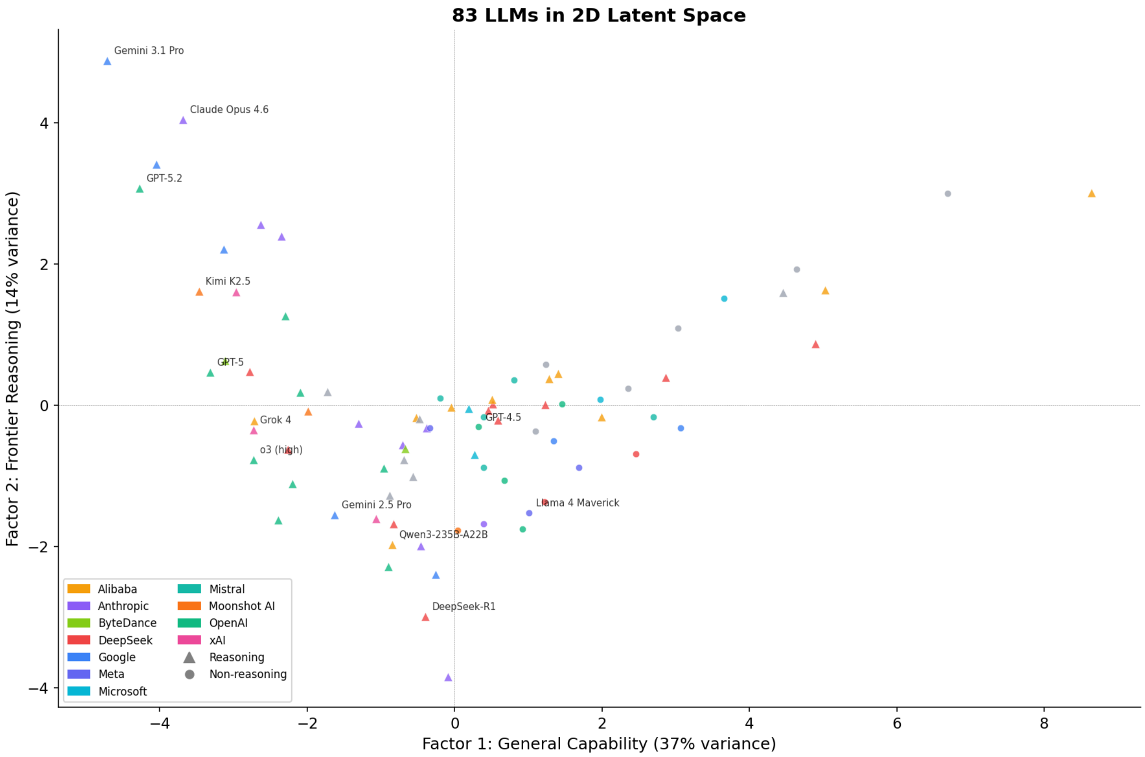

Figure 5: Two-factor latent space. Each point represents a model, colored by provider. Frontier models cluster in the upper-right.

7. Data Efficiency and Phase Transition

How does BenchPress accuracy scale with the number of known benchmark scores per model?

Known Scores	MedAPE	MAE	+/-3 pts	+/-5 pts
1	12.08%	7.9	24.7%	38.9%
2	11.2%	7.0	30.0%	43.6%
3	10.3%	6.3	32.5%	46.1%
5	9.23%	5.6	35.8%	50.0%
7	9.0%	5.4	36.5%	52.0%
10	10.1%	5.9	37.2%	52.5%
15	11.2%	6.8	41.1%	56.8%
20	9.6%	4.5	44.4%	61.1%

The biggest gains come from 1 to 5 scores: MedAPE drops from 12.08% to 9.23%. Each additional benchmark narrows the uncertainty about where the model sits in the latent space. After 5 scores, MedAPE plateaus but the +/-3 and +/-5 point accuracy metrics continue improving.

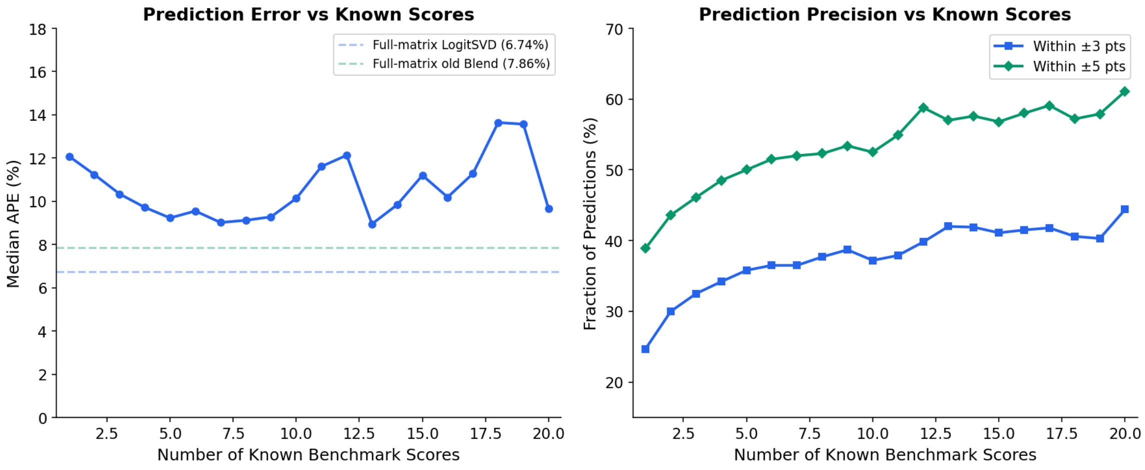

Figure 6: Phase transition. MedAPE as a function of the number of known benchmark scores per model.

7.1 Minimum Evaluation Set

Using greedy forward selection, we identify a 5-benchmark "minimum eval set":

{HLE, AIME 2025, LiveCodeBench, SWE-bench Verified, SimpleQA}

These five benchmarks span four categories (reasoning, math, coding, knowledge) and load on both latent factors. Under proper holdout evaluation (random 20%), this 5-benchmark Ridge predictor achieves ~7.8% MedAPE compared to BenchPress's 5.95%.

GPQA Diamond is a near-perfect substitute for HLE and has 2x the model coverage (81 vs 38 models), making it the pragmatic choice.

8. Scaling Laws and Reasoning Analysis

8.1 Within-Family Scaling

Within model families that vary only in parameter count, benchmark scores scale approximately log-linearly with model size. The DeepSeek distillation family shows remarkably tight scaling (R^2=0.95-0.98), suggesting distillation preserves scaling behavior. The Qwen3 family shows looser fits (R^2=0.77-0.89).

8.2 Reasoning Mode Analysis

Benchmarks with the largest reasoning advantage (z-score gap between 57 reasoning and 26 non-reasoning models):

Benchmark	Reasoning	Non-Reasoning	Gap
HMMT Feb	+0.24	-1.56	+1.80
HLE	+0.18	-1.57	+1.75
AIME 2024	+0.43	-1.13	+1.56
MMMU	+0.40	-1.10	+1.49
ARC-AGI-1	+0.31	-1.12	+1.43

The GSM8K anomaly.

GSM8K is the only benchmark where non-reasoning models outperform reasoning models on average. This occurs because GSM8K is now saturated -- most frontier models score 90%+ -- and the extended reasoning overhead provides no benefit on simple arithmetic word problems.

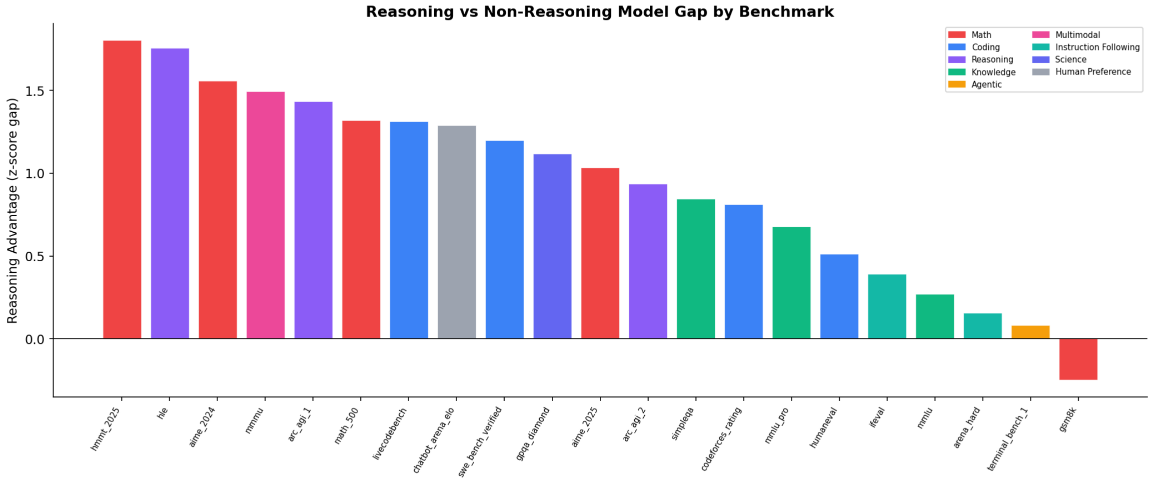

Figure 7: Reasoning mode advantage by benchmark. Positive values indicate benchmarks where reasoning models outperform non-reasoning models.

9. LLMs as Benchmark Predictors

An intriguing alternative to statistical matrix completion is to ask a capable LLM to predict benchmark scores directly.

9.1 Experimental Setup

We format the entire observed matrix as a CSV string (with ? for missing entries) and include it in the system prompt along with structural context. For each batch of ~10 models, the user prompt lists each model's known scores and requests predictions for its missing scores in JSON format.

- Context: Full matrix CSV (~12,600 input tokens per batch)
- Batching: 8 batches of ~10 models each (275 test cells in random 20% holdout)
- Output: JSON with model name -> benchmark -> predicted score
- Total cost: $0.86 (Claude Sonnet 4.5)

9.2 Results (Post-Audit)

Table 4 compares Claude Sonnet 4.5 against the statistical methods on the same random 20% holdout (seed=42, 275 test cells, post-audit matrix).

Method	MedAPE	MAE	+/-3	+/-5	BiAcc	Cov	Cost
BenchPress	5.81%	11.07	40.7%	58.2%	78.6%	100%	$0
Claude Sonnet 4.5	6.08%	12.29	41.5%	56.4%	92.9%	100%	$0.86
BenchReg	6.35%	12.92	37.7%	51.5%	100%	84.0%	$0

Benchmark Mean	11.91%	19.77	12.7%	25.5%	42.9%	100%	$0

9.3 Claude vs. Algorithm: Phase Analysis

The figure below shows what happens when you vary how much the predictor knows. Take Gemini 3.1 Pro, hide all its scores, then reveal them one at a time.

k (known scores)	Algorithm MAE	Claude MAE
0	16.83	2.47
1	12.02	2.93
3	4.93	3.67
5	4.90	4.67
7	3.68	4.93
10	2.73	4.20

At k=0 (zero known scores, just the model's name), Claude already predicts Gemini 3.1 Pro's benchmarks to within 2.47 points. The algorithm at k=0 can only guess column averages and is off by 16.83 points. But by k=5, the algorithm catches up to 4.90 vs Claude's 4.67.

The takeaway: Claude's world knowledge is worth about 5 benchmark scores of information. After that, linear algebra wins.

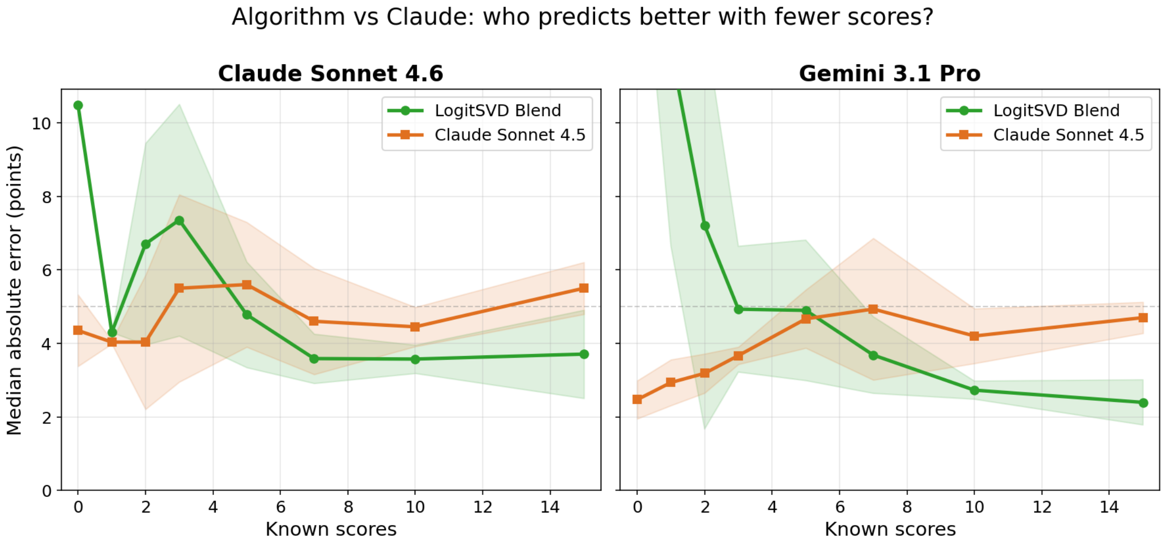

Figure 8: Claude vs. algorithm as a function of known scores. Claude has a massive advantage at k=0 from world knowledge, but the algorithm catches up by k=5.

10. Discussion

10.1 Surprising Models

Some models deviate markedly from low-rank expectations. GPT-4.5 scores only 0.8% on ARC-AGI-2 despite an expected score of ~17% based on its overall profile. Mistral Large 3 scores 44% on GPQA Diamond vs. an expected 67%. These outliers highlight the limits of low-rank approximation.

10.2 Benchmark Redundancy

Within the "Frontier Reasoning" cluster (17 benchmarks including AIME 2025, FrontierMath, HLE, ARC-AGI-2), pairwise correlations exceed 0.6. The "Core Competency" cluster (GPQA Diamond, MMLU-Pro, LiveCodeBench, IFEval) shows similar redundancy. This is both a feature (it enables matrix completion) and a warning (many benchmarks measure overlapping capabilities).

10.3 Limitations

- 33.8% fill rate. The matrix is sparse. Additional scores could be mined from model papers, leaderboards, and community evaluations.
- Blend weight not cross-validated. alpha=0.6 was selected by manual sweep.
- Non-percentage benchmarks (~8%) do not benefit from the logit transform.
- No temporal modeling. The matrix treats all scores as contemporaneous.
- Single evaluation window: January 2025 - February 2026.

11. Related Work

LLM evaluation.

Comprehensive benchmarking efforts include the Open LLM Leaderboard, HELM, and Chatbot Arena. These provide standardized evaluations but do not address the missing data problem we tackle here.

Matrix completion.

The theoretical foundations for low-rank matrix completion were established by Candes and Recht (2009) and Candes and Tao (2010). Soft-impute SVD (Mazumder et al., 2010) provides the iterative algorithm we use. Our contribution is the application of logit-space transforms to benchmark score completion.

Benchmark prediction.

Concurrent work on predicting benchmark scores includes scaling law extrapolation (Kaplan et al., 2020) and benchmark-specific forecasting. Our approach differs in treating the cross-benchmark prediction problem: given scores on some benchmarks, predict scores on others.

12. Conclusion and Future Work

We have constructed a fully-cited benchmark matrix of 83 LLMs across 49 evaluations and developed BenchPress, a matrix completion method achieving 7.25% median error on per-model leave-50%-out holdout (3 seeds, post-audit). The key insight is that applying the logit transform before regression and factorization linearizes ceiling/floor effects and implicitly handles bimodal benchmarks. The matrix exhibits strong low-rank structure, with the first principal component of a complete submatrix explaining 71.3% of the variance.

We also demonstrate that Claude Sonnet 4.5 can predict benchmark scores competitively (6.08% MedAPE on random holdout) when given the matrix as context at a cost of $0.86. This raises the intriguing possibility that LLMs encode implicit representations of model capabilities that can be leveraged for evaluation.

Future work.

- Active learning: Choosing which benchmark to evaluate next for maximum information gain.

- Temporal modeling: Incorporating benchmark vintage and saturation curves.

- Confidence intervals: Predicting uncertainty bounds alongside point estimates.

- Larger matrices: Expanding to more models, benchmarks, and evaluation conditions.

- Hybrid approaches: Combining LLM predictions with statistical methods.

Reproducibility: All code, data, and results are publicly available at github.com/anadim/llm-benchmark-matrix.

. References

[1] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," Foundations of Computational Mathematics, vol. 9, no. 6, pp. 717-772, 2009.

[2] E. J. Candes and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," IEEE Transactions on Information Theory, vol. 56, no. 5, pp. 2053-2080, 2010.

[3] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," JMLR, vol. 11, pp. 2287-2322, 2010.

[4] J. Kaplan et al., "Scaling laws for neural language models," arXiv:2001.08361, 2020.

[5] E. Beeching et al., "Open LLM Leaderboard," Hugging Face, 2023-2026.

[6] P. Liang et al., "Holistic Evaluation of Language Models," TMLR, 2023.

[7] W.-L. Chiang et al., "Chatbot Arena: An open platform for evaluating LLMs by human preference," arXiv:2403.04132, 2024.