
Your Benchmark Is an Empirical Measure Over Difficulty

Yifan Sun^{†*} Naicheng Yu^{‡*} Jiawen Gong^{†*} Jingyan Shen[§] Huan Zhang[†]

[†]University of Illinois Urbana-Champaign [‡]University of California San Diego

[§]New York University

Abstract

A benchmark is not merely a collection of test questions; it naturally induces an empirical measure over difficulty. In this work, we develop this measure-based view of benchmark evaluation. This perspective allows us to identify and formalize two central failure modes: *measure mismatch*, where the benchmark difficulty distribution differs from the intended difficulty distribution, and *coverage gaps*, where uncovered regions along the difficulty axis prevent benchmarks from detecting meaningful localized training progress. We propose two diagnostic metrics that capture these two failure modes and provide rigorous theoretical guarantees. Empirically, we analyze **25** popular benchmarks spanning **7** domains using carefully estimated item difficulties. Our results show that many benchmarks implicitly overweight certain difficulty regions, so their leaderboard conclusions should be interpreted with caution; others leave substantial portions of the difficulty axis uncovered, limiting their ability to monitor the model improvement process. Finally, we turn diagnosis into action by introducing BENCHFILL, a simple difficulty-targeted item rewriting algorithm that effectively reduces local coverage gaps.

1 Introduction

Benchmark evaluation looks simple: we run models on a set of questions and average the results. The resulting score is then used in two main ways: to compare models on leaderboards [2, 61, 29, 45], and to monitor progress during training [33, 21, 16, 6, 38]. But this simplicity hides a basic issue: the conclusion depends not only on how well models perform, but also on how benchmark questions are distributed [58, 68].

Consider first the leaderboard setting in Fig. 1(a), where we want to compare two models. Suppose they perform similarly on easy questions but differ substantially on hard ones. If the benchmark places most of its weight on easy questions, while the difficulty distribution of interest places more weight on harder questions, the two evaluations can lead to different conclusions for the same pair of models: the benchmark may suggest that the two models are nearly tied, even though under the desired difficulty distribution they are clearly separated. Now consider the training setting in Fig. 1(b), where we want to track whether a model is improving. Suppose the model improves *substantially* over a local difficulty range. If the benchmark has few or no questions in that region, the benchmark score may barely change, even though the model has made real progress.

These two failure cases suggest that benchmark conclusions depend critically on how questions are distributed. To study this structure systematically, we organize benchmark questions along a **difficulty** axis, which provides a natural progression from easier to harder problems. The key conceptual step is to use difficulty as a way to define the distributional structure of the benchmark itself. This view

*Equal contribution.

Correspondence to Yifan Sun <yifan50@illinois.edu> and Huan Zhang <huan@huan-zhang.com>.

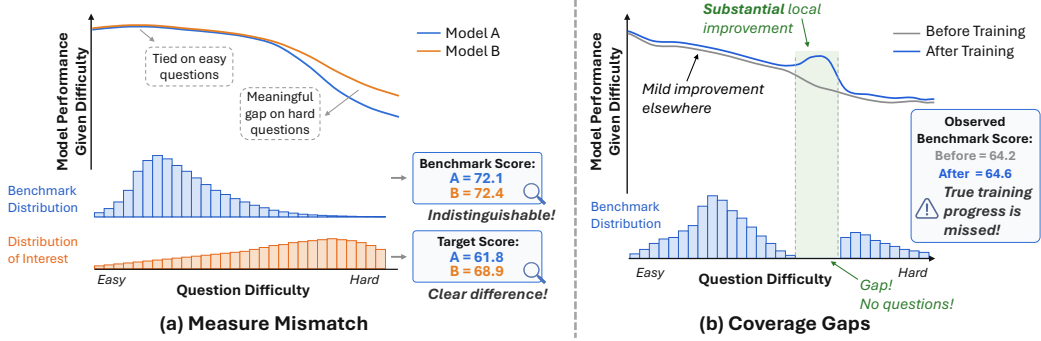


Figure 1: **Two benchmark failure cases.** In both panels, the curves show true model *conditional* performance *as a function of question difficulty*. (a) **Measure Mismatch.** The benchmark emphasizes easy questions, where the models perform similarly, while the difficulty distribution we care about places more weight on harder questions, where they differ substantially. Because the benchmark and the target distribution weight the difficulty axis differently, they lead to different conclusions for the same two models. (b) **Coverage Gaps.** A model actually improves *substantially* over a localized difficulty region. Because the benchmark has no questions in that region, this improvement is not reflected in the observed benchmark score, causing meaningful training progress to be missed.

recasts a benchmark from a collection of test questions into an **empirical measure** over difficulty, with each question contributing equal evaluation mass at its difficulty level.

This measure-theoretic formalization is useful because it turns the two failure cases above into precise and analyzable problems about the benchmark-induced difficulty measure. The first is a **measure mismatch** problem: the benchmark may distribute its mass differently from the target difficulty distribution that represents the intended evaluation objective, so benchmark conclusions can differ from the conclusions under the target difficulty distribution. The second is a **coverage gap** problem. Because the benchmark is an empirical measure supported only on finitely many difficulty points, there may be intervals of difficulty with little or no benchmark mass. A substantial performance gain concentrated in such an interval may therefore produce little or no change in the benchmark score. To our knowledge, this is the first work to identify and formalize measure mismatch and coverage gaps as two fundamental failure modes of benchmark evaluation.

These two problems naturally lead to two simple diagnostic metrics: the Wasserstein distance [60] between the benchmark measure and a target difficulty distribution, and the maximum gap between adjacent benchmark difficulty points. We provide theoretical results showing that these quantities directly characterize distributional mismatch and the detectability of localized improvements. Empirically, we analyze **25** popular benchmarks spanning **7** domains. We estimate question difficulty using Item Response Theory and compute the proposed metrics. Our results show that both issues are prevalent across benchmarks.

Our key contributions are summarized as follows:

- We introduce a new measure-based perspective on benchmark evaluation, representing a benchmark as an empirical measure over difficulty. This perspective formalizes measure mismatch and coverage gaps, motivates two diagnostic metrics, and yields rigorous theoretical guarantees.
- We conduct an empirical study across **25** benchmarks spanning **7** domains, showing that both difficulty-distribution mismatch and large coverage gaps are widespread in current benchmarks.
- We propose BENCHFILL, a simple difficulty-targeted gap-filling algorithm. With only three inserted items per benchmark, BENCHFILL reduces the sum of the top three difficulty gaps by 21.09%–38.88%.

2 Formalization and Theory

Difficulty space and model performance function. We represent the difficulty of a benchmark item, e.g., a question, by a continuous variable $z \in [0, 1]$, where larger values correspond to harder

items. For a model θ , let $s_\theta(z) \in [0, 1]$ denote its conditional expected performance on items with difficulty z , e.g., pass@1.

Benchmark distribution and score. Let a benchmark $B = \{x_i\}_{i=1}^n$ consist of n items, and let $z_i \in [0, 1]$ denote the difficulty of item x_i . The benchmark score is $S_B(\theta) := \frac{1}{n} \sum_{i=1}^n s_\theta(z_i)$. This average implicitly induces an empirical measure over the difficulty axis,

$$\nu_B = \frac{1}{n} \sum_{i=1}^n \delta_{z_i},$$

where δ_z denotes a point mass at difficulty z . Under this representation, the benchmark score is exactly an integral with respect to the benchmark-induced difficulty measure, $S_B(\theta) = \int s_\theta(z) d\nu_B(z)$.

Performance difference between two models. For two models θ and θ' , define the *difficulty-conditioned performance difference* as $\Delta(z) := s_{\theta'}(z) - s_\theta(z)$, where the dependence of Δ on (θ', θ) is omitted for notational simplicity. The performance difference *observed on benchmark B* is

$$D_B(\theta', \theta) := S_B(\theta') - S_B(\theta) = \frac{1}{n} \sum_{i=1}^n \Delta(z_i) = \int \Delta(z) d\nu_B(z).$$

2.1 Measure Mismatch

Benchmark-induced coupling. The quantity $D_B(\theta', \theta)$ implicitly couples two distinct objects: the *model-dependent function* $\Delta(z)$ and the *benchmark-dependent measure* ν_B . Thus, a leaderboard comparison is not merely determined by the difficulty-conditioned performance difference between the models; it also depends on how the benchmark allocates mass over the difficulty axis. If this allocation differs from the difficulty distribution of interest, the benchmark aggregates the same model difference under a mismatched measure. This mismatch can distort the observed model gap and, in severe cases, change the induced ordering.

Target distribution. Formally, we introduce a target distribution $\mu \in \mathcal{P}([0, 1])$, which represents the intended evaluation distribution over the difficulty space. This target distribution may encode the desired application mix. Under μ , the target score of model θ and the target performance difference between θ' and θ are defined as

$$S_\mu(\theta) := \int s_\theta(z) d\mu(z), \quad D_\mu(\theta', \theta) := S_\mu(\theta') - S_\mu(\theta) = \int \Delta(z) d\mu(z).$$

The discrepancy between $D_B(\theta', \theta)$ and $D_\mu(\theta', \theta)$ arises from the mismatch between the benchmark measure ν_B and the target distribution μ . This viewpoint allows us to study distortions in leaderboard comparisons as a distribution mismatch problem.

A metric for distribution mismatch. We measure how well the difficulty distribution of a benchmark matches a target difficulty distribution using the 1-Wasserstein distance. Specifically, for a benchmark B with empirical difficulty measure ν_B and a target distribution μ on $[0, 1]$, we define its *Difficulty Distribution Mismatch (DDM)* as

$$\text{DDM}(B; \mu) := W_1(\nu_B, \mu) = \inf_{\pi \in \Pi(\nu_B, \mu)} \int_{[0, 1]^2} |u - v| d\pi(u, v),$$

where $\Pi(\nu_B, \mu)$ denotes the set of all couplings of ν_B and μ . Intuitively, $W_1(\nu_B, \mu)$ measures how much probability mass must be transported, and how far, to transform the benchmark difficulty distribution into the target one. The following theorem shows that this mismatch directly controls the distortion between benchmark-based and target-based performance differences.

Theorem 2.1 (Distribution mismatch controls score distortion and rank stability). *Suppose the performance difference function Δ is L -Lipschitz. Then*

$$|D_B(\theta', \theta) - D_\mu(\theta', \theta)| \leq L W_1(\nu_B, \mu).$$

Consequently, if $|D_\mu(\theta', \theta)| > L W_1(\nu_B, \mu)$, then

$$\text{sign}(D_B(\theta', \theta)) = \text{sign}(D_\mu(\theta', \theta)),$$

so the benchmark-induced ordering agrees with the target ordering, i.e., no rank flip can occur.

Theorem 2.1 shows that $W_1(\nu_B, \mu)$ bounds how much the benchmark-induced score difference can differ from the target score difference. A small $W_1(\nu_B, \mu)$ means that the benchmark provides a faithful approximation to the target evaluation, while a large $W_1(\nu_B, \mu)$ indicates that the benchmark may overemphasize some difficulty regions and underrepresent others, potentially distorting model comparisons and even changing their induced ranking. App. B.1 provides the full proof and extends this result to the finite-sample setting.

A principled target distribution without prior information. When prior information is available, it is natural to choose μ to reflect the intended evaluation objective over the difficulty space. In many settings, however, such prior knowledge is unavailable. In that case, choosing a non-uniform target would implicitly privilege some difficulty regions over others without evidence. To characterize a principled default choice, we consider an agnostic setting in which the location of the performance difference along the difficulty axis is completely unknown. We consider the class

$$\mathcal{F}_\rho := \left\{ \Delta : [0, 1] \rightarrow [-1, 1] \text{ measurable} : \int_0^1 \Delta(z) dz = \rho \right\}, \quad \rho \in (-1, 1).$$

This class encodes an uncertainty set: the signed performance difference has a fixed average level ρ over the difficulty axis, but its location is otherwise unrestricted.

For any candidate target distribution $\mu \in \mathcal{P}([0, 1])$, define its worst-case retained difference by

$$V_\rho(\mu) := \inf_{\Delta \in \mathcal{F}_\rho} \int \Delta(z) d\mu(z).$$

The following theorem shows that, under complete uncertainty about where along the difficulty axis the performance difference occurs, the uniform distribution is the *unique* maximin target, and is the only choice whose worst-case retained difference remains equal to the average difference ρ .

Theorem 2.2 (Uniform target as the unique maximin choice). *For $\rho \in (-1, 1)$, we have*

$$\sup_{\mu \in \mathcal{P}([0, 1])} V_\rho(\mu) = \rho,$$

and the unique maximizer is the uniform distribution $\lambda = \text{Unif}[0, 1]$.

Theorem 2.2 does not claim that the uniform distribution is universally optimal. Rather, it justifies the uniform target as a principled default when no prior information over the difficulty space is available. In that case, adopting the uniform target avoids introducing an additional preference over difficulty regions and provides a neutrality criterion for benchmark evaluation: the evaluation does not privilege any difficulty region a priori. Full proof is provided in App. B.1.

2.2 Coverage Gaps

Discrete benchmark measures may fail to monitor local training improvements. Benchmarks are often used not only for final model comparison, but also to monitor training progress [16, 65, 38]. However, training improvements are often local rather than uniform: a model may improve substantially over a particular difficulty range while improving only modestly elsewhere.

Recall that a benchmark score is an integral with respect to the benchmark empirical measure ν_B . Since ν_B is a *discrete* measure supported only on the finitely many benchmark difficulty points, it assigns zero mass to every interval that contains no benchmark item. Consequently, even if the model improves substantially on a nontrivial contiguous difficulty region, the benchmark score may not reflect this improvement when the region contains no benchmark item. In this sense, a benchmark can fail as a progress monitor when its local resolution is insufficient.

Maximum gap. We first characterize the local resolution of a benchmark through its *maximum gap*. Let $0 \leq z_{(1)} \leq \dots \leq z_{(n)} \leq 1$ be the ordered benchmark difficulty points, and define $z_{(0)} := 0$ and $z_{(n+1)} := 1$. The *maximum gap* of benchmark B is

$$G(B) := \max_{0 \leq j \leq n} (z_{(j+1)} - z_{(j)}) \tag{1}$$

i.e., the length of the largest interval on the difficulty axis that contains no benchmark point.

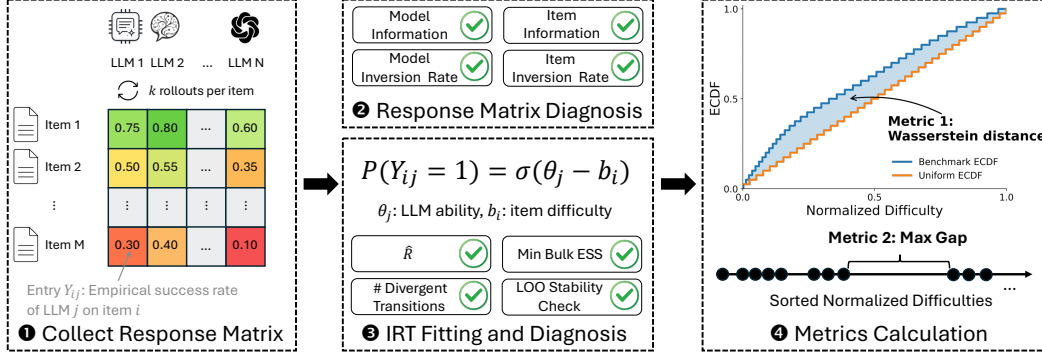


Figure 2: **Overview of our benchmark difficulty estimation and metric calculation pipeline.** We collect empirical success rates from multiple LLMs on benchmark items, diagnose the response matrix, fit a IPL IRT model to estimate item difficulties, perform post-fit IRT diagnostics, and compute two metrics over the normalized difficulty distribution.

A hypothesis testing formulation. To formalize whether a benchmark can reliably detect localized training improvements, we cast the problem as a hypothesis test. Under the null hypothesis, there is no improvement. Under the alternative, the post-improvement performance exceeds the pre-improvement performance by at least a over some contiguous difficulty interval of width w . The following theorem shows that the benchmark maximum gap $G(B)$ is the exact asymptotic minimax detection boundary.

Theorem 2.3 (Maximum gap as an exact asymptotic minimax detection boundary). *Fix a test level $\alpha \in (0, 1)$ and $a \in (0, 1]$. Let $s^-, s^+ : [0, 1] \rightarrow [0, 1]$ denote the pre- and post-improvement performance curves, respectively. For a benchmark B , let $\Pi_\alpha^*(B; w, a, k)$ denote the minimax worst-case power of level- α tests based on k repeated samples per benchmark item for testing*

$$H_0 : s^- = s^+ \quad \text{versus} \quad H_1 : \exists \text{ interval } I \subset [0, 1], |I| = w, s^+(z) - s^-(z) \geq a \quad \forall z \in I.$$

1. If $w \leq G(B)$, then $\Pi_\alpha^*(B; w, a, k) = \alpha$ for every $k \in \mathbb{N}$.
2. If $w > G(B)$, then $\lim_{k \rightarrow \infty} \Pi_\alpha^*(B; w, a, k) = 1$.

Therefore, $G(B)$ exactly characterizes the asymptotic minimax detection boundary for benchmark B .

Theorem 2.3 shows that the discrete geometry of the benchmark induces a sharp detectability phase transition. If the width of a local improvement does not exceed the maximum adjacent difficulty gap, then no level- α test can achieve nontrivial worst-case power, regardless of how many repeated evaluations are performed. In other words, such an improvement can be completely invisible to the benchmark in the worst case. Once the improvement width exceeds the maximum gap, repeated evaluation can eventually make the signal detectable. In this sense, $G(B)$ characterizes the intrinsic local resolution limit of benchmark B . The full proof is given in App. B.2.

3 Empirical Diagnosis of Current Benchmarks

We analyze a collection of widely used benchmarks from diverse domains using the two metrics proposed earlier. Fig. 2 provides an overview of our difficulty estimation and metric calculation pipeline. Specifically, we consider **25** popular benchmarks spanning math, coding, agentic tasks, logic, science, instruction-following, and medicine. Detailed information about these benchmarks is provided in App. C.1.

3.1 Difficulty Estimation and Metrics Calculation

Method: Item Response Theory. To estimate benchmark item difficulty, one may be tempted to use each item’s average failure rate across models. However, item difficulty is inherently relative to model ability: it should depend not only on how many models fail an item, but also on how capable the models that fail or solve it are. We therefore use Item Response Theory (IRT), an established psychometric framework [35, 49] that has been widely adopted for difficulty estimation in LLM benchmark analysis [12, 52, 20, 25, 47, 69]. IRT models the probability that a model solves an item

as a function of both model ability and item difficulty. In the commonly used one-parameter logistic model (1PL), each model j is assigned an ability parameter θ_j , and each item i is assigned a difficulty parameter b_i . The probability that model j correctly solves item i is modeled as

$$\Pr(Y_{ij} = 1) = \sigma(\theta_j - b_i),$$

where $\sigma(\cdot)$ is the logistic function. Intuitively, items with larger b_i require higher model ability to be solved correctly, and the estimated b_i therefore provides a natural measure of item difficulty. For multiple-choice benchmarks, the model above can be naturally extended with a guessing parameter.

In practice, difficulty estimation begins by collecting the performance of multiple models on benchmark items, forming a response matrix $Y \in \mathbb{R}^{n \times m}$, where Y_{ij} denotes the empirical success rate of model j on item i . The latent abilities $\{\theta_j\}_{j=1}^m$ and item difficulties $\{b_i\}_{i=1}^n$ are then estimated by fitting the likelihood induced by the IRT model. To ensure reliable difficulty estimation, we take several practical measures, including **careful model selection** and **rigorous IRT fitting diagnostics**.

Model selection. Accurate difficulty estimation requires models with diverse capability levels. To ensure sufficient ability coverage, we deliberately select models spanning a wide range of capabilities. For non-agentic benchmarks, we select 12 LLMs, ranging from 1.7B to 397B parameters, using the public leaderboard from Text Arena¹. Models are first ordered by their overall capability aggregated across multiple domains, then divided into several approximately equal intervals along this ranking. We perform stratified sampling by selecting one model from each interval and additionally include a few models from both the top and bottom of the ranking to better capture the extremes of the capability spectrum. Each selected model is evaluated on each benchmark item with up to 32 rollouts using the model-recommended temperature and top- p . For agentic benchmarks, tasks typically require strong proprietary models. We therefore rely on the evaluation results from HAL [24], which reports the performance of 15–66 state-of-the-art closed-source LLMs on agentic task benchmarks. Detailed information about the models is provided in App. C.2.

IRT fitting and diagnostics. We estimate the IRT parameters using Bayesian inference via MCMC. To ensure reliable difficulty estimation, we perform a series of diagnostics both *before* and *after* IRT fitting. Before fitting, we examine the informativeness and structural consistency of the response matrix using 4 metrics. After fitting, we assess convergence and sampling quality using 3 standard Bayesian diagnostics, including the Gelman–Rubin statistic ($\hat{R}$), effective sample size (ESS), and the number of divergent transitions. All benchmarks pass all 7 diagnostic checks. For non-agentic benchmarks, we further conduct a leave-one-out (LOO) robustness analysis. Results suggest that our choice of 12 models is sufficient for stable difficulty estimation. To support future work, we will open-source the carefully curated difficulty estimates for the 25 benchmarks. Detailed definitions and diagnostic results are provided in App. C.3. We also provide additional model-comparison analyses against 2PL and multidimensional IRT models in App. C.5.

Proposed metric calculation. For each benchmark, after estimating item difficulties, we first normalize them to $[0, 1]$ and compute their empirical cumulative distribution function (ECDF). Since in general no additional prior information over the difficulty space is available, we quantify distribution mismatch by computing the 1-Wasserstein distance between the benchmark empirical measure and a same-size empirical *uniform reference* on $[0, 1]$. In one dimension, this is equivalently the L_1 distance between their ECDFs. We compute the maximum gap according to Eq. (1) and also report the sum of the five largest adjacent gaps as a reference metric. We provide posterior uncertainty analyses for both metrics in App. C.6.

3.2 Results and Analysis

Several benchmarks exhibit large deviations from the neutral uniform target. Tab. 1 shows that 10 out of 25 benchmarks have $\text{DDM} \geq 0.10$, indicating that their empirical difficulty distributions must be transported by an average distance of at least 10% of the full difficulty range to match the same-size uniform reference. These benchmarks therefore encode a strong implicit difficulty weighting: their aggregate scores are more sensitive to overrepresented difficulty regions and less sensitive to underrepresented ones. Non-uniformity is not inherently problematic when it matches the

¹<https://arena.ai/leaderboard>

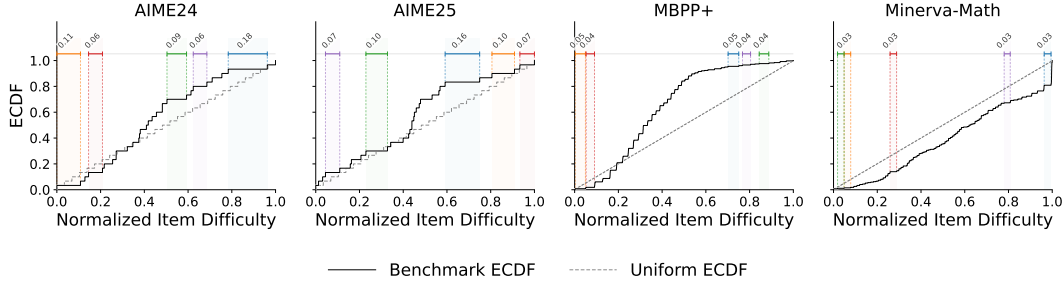


Figure 3: **Benchmark difficulty distributions.** Each panel shows the ECDF of normalized item difficulties for one benchmark, together with the neutral uniform reference. Colored shaded intervals mark the five largest adjacent difficulty gaps, with labels denoting the corresponding gap sizes. Additional benchmark examples are provided in Fig. 5.

Table 1: **Summary of proposed metrics across benchmarks.** DDM denotes the 1-Wasserstein distance between the benchmark empirical difficulty distribution and a same-size empirical uniform distribution on $[0, 1]$. Max Gap is the largest gap between consecutive normalized difficulties, and Top-5 Sum is the sum of the five largest such gaps. # denotes the number of benchmark items. Lower is better for all three metrics. Highlighted entries mark comparatively large values for each metric: $DDM \geq 0.10$, $Max\ Gap \geq 0.10$, or $Top-5\ Sum \geq 0.30$.

Benchmark	DDM	Max Gap	Top-5 Sum	#	Benchmark	DDM	Max Gap	Top-5 Sum	#
MATH-500	0.164	0.058	0.175	500	HumanEval	0.148	0.165	0.423	164
Minerva-Math	0.122	0.031	0.149	272	MBPP+	0.163	0.049	0.214	378
OlympiadBench	0.064	0.048	0.142	675	GAIA	0.084	0.051	0.218	165
Gaokao2023en	0.088	0.055	0.183	385	CORE-Bench Hard	0.036	0.113	0.363	45
AMC23	0.088	0.175	0.528	40	SWE-bench-Mini	0.053	0.149	0.469	50
AIME24	0.077	0.179	0.502	30	USACO	0.080	0.082	0.354	307
AIME25	0.079	0.158	0.492	30	IFEval	0.159	0.046	0.190	541
AIME26	0.085	0.119	0.507	30	BBH-3 objects	0.090	0.088	0.354	250
MMLU-Pro-Biology	0.125	0.043	0.176	717	BBH-5 objects	0.099	0.128	0.332	250
MMLU-Pro-Chemistry	0.114	0.072	0.192	1132	BBH-7 objects	0.077	0.059	0.279	250
MMLU-Pro-Physics	0.121	0.035	0.157	1299	ZebraLogic	0.080	0.027	0.127	1000
GPQA-Diamond	0.080	0.066	0.263	198	MedQA	0.157	0.042	0.178	1273
ScienceQA	0.178	0.045	0.175	4241					

intended evaluation objective. However, when no such objective is specified, high-DDM leaderboard results should not be interpreted as difficulty-neutral comparisons. Instead, they reflect comparisons under the benchmark’s empirical difficulty weighting, so users should check whether this weighting matches the difficulty distribution relevant to their use case. App. D presents a simple benchmark calibration algorithm for moving a benchmark distribution closer to a chosen target.

Many benchmarks exhibit large local coverage gaps. Tab. 1 shows that 8 out of 25 benchmarks have $Max\ Gap \geq 0.10$. As a result, a training run may make meaningful local progress in an uncovered difficulty region, while the aggregate benchmark score remains nearly unchanged. This can cause useful training progress to be underestimated, promising runs to be prematurely discarded, and compute to be misallocated. Moreover, the problem is not limited to the single largest gap. We find that 10 out of 25 benchmarks have $Top-5\ Sum \geq 0.30$. For example, Fig. 3 shows that AIME24 and AIME25 have top-5 gap sums close to 0.5, leaving large portions of the difficulty axis locally unresolved despite their widespread use for monitoring RLVR training [38, 33, 14, 66]. Finally, we note that coverage gaps are not merely a small-benchmark artifact²: benchmarks with hundreds of items, such as HumanEval and BBH-5 objects, also exhibit substantial gap metrics. This motivates gap filling as a practical and necessary way to improve local benchmark resolution.

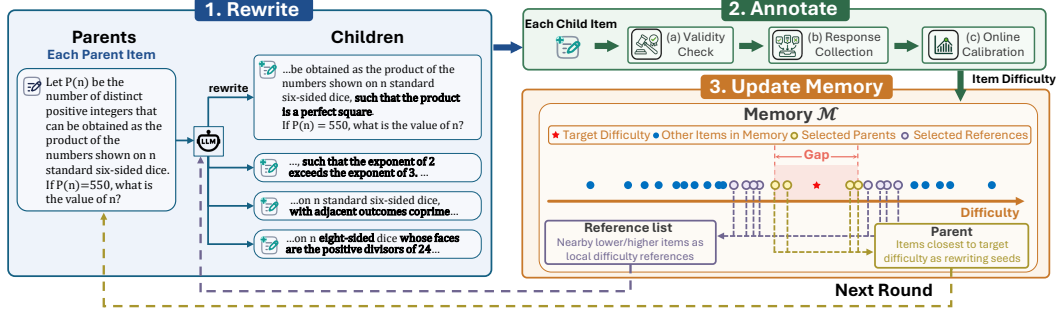


Figure 4: **Overview of BENCHFILL.** BENCHFILL iteratively rewrites parent items into child variants, annotates their difficulties, and updates the memory to guide the next round of gap filling.

4 BENCHFILL: Difficulty-Targeted Gap Filling

4.1 Problem Setting and Method

Problem setting. Let $B = \{x_i\}_{i=1}^n$ be the current benchmark, where each item x_i has an estimated difficulty z_i . We sort the difficulties and consider an adjacent gap $[z_L, z_R]$ between two benchmark items. Reducing such gaps improves a benchmark’s local resolution and its ability to capture localized model improvements. Thus, our goal is to obtain a valid new item x_{new} whose difficulty is as close as possible to the target difficulty $z^* = (z_L + z_R)/2$, the midpoint that most reduces the gap under a single insertion. The new item is added to the benchmark rather than replacing any existing item. By default, we apply this procedure to the *largest* gap of the benchmark.

One direct approach is to ask human experts to design new items at the target difficulty, but this process can be costly [50, 46]. Motivated by recent work on LLM-assisted benchmark augmentation [10, 28, 64, 63], we instead use LLMs to *rewrite* existing benchmark items into new variants at targeted difficulty levels. This rewriting-based approach is scalable, uses existing items as local references for controlled difficulty adjustment, and helps preserve the format and style of the original benchmark. To ensure item quality, the final selected items are further validated by human experts. We next introduce BENCHFILL, a rewrite-and-annotate evolutionary search algorithm [53, 34] for automated difficulty-targeted item rewriting. Alg. 2 gives the pseudocode, and Fig. 4 illustrates the procedure.

BENCHFILL. Given a target difficulty z^* , our algorithm maintains a memory $\mathcal{M}$ of annotated item–difficulty pairs. The memory is initialized with existing benchmark items whose difficulties are closest to the selected gap, and is later updated during the search. Each round follows a parent–child rewriting step. A *parent item* is an existing or previously rewritten item selected from $\mathcal{M}$ as the main reference for rewriting, while a *child item* is a newly proposed variant derived from a parent.

At round t , we retrieve a small set of parent items from $\mathcal{M}$ whose annotated difficulties are closest to the target difficulty z^* . For each parent item, the rewriter LLM is conditioned on the parent, the target difficulty, and a reference list of nearby item–difficulty pairs. The reference list provides local examples around z^* , typically including items slightly below and above the target difficulty, e.g., items with difficulties 0.48, 0.50, 0.60, and 0.62 when $z^* = 0.55$. The rewriter then rewrites each parent into a batch of child items while preserving the format and style of the benchmark.

After rewriting, each child item is passed to the difficulty annotator module, detailed below, which returns an estimated difficulty $\hat{z}$. We keep only valid³ child items and add their annotated item–difficulty pairs $(x, \hat{z})$ back to $\mathcal{M}$. This update makes the search adaptive: whenever a rewritten child is closer to z^* than previous items, it becomes more likely to be retrieved as a parent or included in the reference list in later rounds. The rewriter is therefore conditioned on increasingly target-relevant examples as the search progresses, rather than rewriting each batch independently. After T rounds,

²Although Max Gap intrinsically depends on benchmark size, this reflects a real coverage limitation rather than merely a nuisance effect: even if a large gap is expected for a small benchmark, it is still an uncovered difficulty interval, and by Theorem 2.3, localized improvements inside such an interval may be undetectable in the worst case.

³We use GPT-5.4 as a validity judge to verify whether the answer associated with each rewritten item is correct.

Table 2: **Gap-filling results.** Top-3 Gap Sum Reduction reports the relative reduction (%) from the original benchmark, where larger is better. Target Dev. is the mean absolute deviation between the selected item difficulty and the target difficulty over three filled gaps, where lower is better. Baseline uses a budget-matched best-of- N uniform-parent rewriting strategy. Oracle is an *idealized upper bound* that inserts an item exactly at each target difficulty, rather than an actual gap-filling method.

Benchmark	Top-3 Gap Sum Reduction (%) $\uparrow$			Target Dev. $\downarrow$	
	Baseline	BENCHFILL	Oracle	Baseline	BENCHFILL
AIME24	6.90	35.57	35.57	0.057	0.005
AIME25	27.44	37.56	37.56	0.016	0.005
AIME26	17.94	33.04	33.04	0.045	0.002
AMC23	24.47	35.19	35.78	0.009	0.003
Gaokao2023en	11.45	38.88	45.47	0.152	0.004
BBH-3 objects	14.83	29.88	32.80	0.026	0.006
MMLU-Pro-Biology	22.23	30.13	36.84	0.046	0.006
IFEval	14.94	21.09	23.44	0.036	0.007
Average	17.52	32.67	35.06	0.048	0.005

we select the valid child item minimizing $|\hat{z} - z^*|$ as the proposed gap-filling item. Before insertion into the benchmark, the final item is subject to careful manual review by human experts.

Difficulty annotation module details. For each rewritten item x , we annotate its difficulty by online calibration under the IRT model fitted on the original benchmark. Let $\hat{\theta}_j$ be the pre-estimated ability of model j . We evaluate each model on x for K_j trials and observe c_j correct responses. The item difficulty is estimated by fitting only the new item parameter:

$$\hat{b}(x) = \arg \max_b \sum_{j=1}^m \left[c_j \log \sigma(\hat{\theta}_j - b) + (K_j - c_j) \log(1 - \sigma(\hat{\theta}_j - b)) \right],$$

where $\sigma(\cdot)$ is the logistic function⁴. This online calibration places each rewritten item on the same IRT scale as the original benchmark without refitting the full model.

4.2 Experimental Setup and Results

Experimental setup. We perform gap filling on 8 representative benchmarks from diverse domains. For each benchmark, we target the three most pronounced coverage gaps. We use MiniMax-M2.7 [7] as the rewriter LLM, with temperature set to 1.0. The search runs for at most 8 rounds; in each round, we retrieve 4 parent items, rewrite each parent into 4 child items, and provide a reference list of at most 16 nearby item–difficulty pairs. As a baseline, we use a budget-matched best-of- N uniform-parent rewriting variant. Additional implementation details are provided in App. E.1.

Results. As shown in Tab. 2, with only three inserted items per benchmark, BENCHFILL reduces Top-3 Sum by 21.09%–38.88%. We also include the oracle result, but only as a reference upper bound, not as an implementable gap-filling method. The achieved reductions are close to this oracle reference: on average, BENCHFILL achieves a 32.67% reduction, compared with 35.06% for the oracle and 17.52% for the budget-matched uniform-parent baseline. On AIME24, AIME25, and AIME26, BENCHFILL reaches the oracle reduction. These reductions show that BENCHFILL improves coverage of the largest under-covered difficulty regions, which can strengthen a benchmark’s ability to monitor localized training progress. We further report Target Dev. and BENCHFILL consistently achieves a Target Dev. below 0.01, indicating that the rewrite-and-annotate search can produce items close to the intended difficulty levels. Representative generated items are shown in Tab. 13. We will release the gap-filling benchmark items to support future evaluation studies.

5 Related Work

Rethinking benchmark analysis. A recent line of work has shifted attention from improving leaderboard scores to examining whether benchmark scores themselves provide reliable scientific

⁴For multiple-choice benchmarks, we additionally incorporate the same chance-level guessing parameter used in the original IRT fitting, which is omitted here for notational simplicity.

evidence. Prior studies show that benchmark results can be sensitive to decoding, prompting, and implementation details [19], and brittle to surface wording or dataset flaws [44]. Other work studies evaluation reliability through the lens of signal and noise [17], or proposes broader benchmark-quality criteria and shows that widely used benchmarks vary substantially in quality [48]. Our work complements these efforts by formulating a benchmark as an empirical measure over difficulty, enabling us to rigorously formalize measure mismatch and coverage gaps.

IRT and its applications. IRT originated in psychometrics as a framework for educational and psychological measurement [13, 36]. In NLP, [26] introduced IRT to system evaluation, and [52] extended this perspective to leaderboard analysis. IRT has since become a widely used foundation for difficulty-aware evaluation in LLM benchmarking. Easy2Hard-Bench [12] uses IRT to assign standardized continuous difficulty labels to benchmark items; it also validates these labels against human pairwise judgments and reports stronger alignment than GPT-4-based difficulty annotators. [69] uses an IRT-based framework to analyze the measurement quality of existing LLM benchmarks.

6 Conclusion

In this work, we view each benchmark as an empirical measure over difficulty. This perspective enables us to formalize measure mismatch and coverage gaps, leading to theoretical results and practical diagnostics. We believe this empirical-measure view has the potential to open new ways to rigorously study and rethink benchmark evaluation.

7 Limitations and Future Work

This work develops a measure-theoretic view of benchmark evaluation and uses it to characterize two central issues: measure mismatch and coverage gaps. Our goal is not to exhaust all consequences of this viewpoint, but to show that benchmark evaluation can be studied through the geometry of the empirical difficulty measure induced by the benchmark. We believe this perspective opens a broader direction for studying benchmark evaluation, where additional benchmark failures and design objectives can be formulated as properties of the induced difficulty measure.

Moreover, our current framework is primarily post-hoc: given an existing benchmark, we estimate its difficulty distribution, diagnose measure mismatch and coverage gaps, and reduce pronounced gaps through targeted item insertion. A natural next step is to move this perspective into the benchmark construction stage itself. Rather than auditing mismatch and coverage only after a benchmark has been built, benchmark developers can actively design the induced difficulty structure from the beginning, aiming to align the benchmark measure with the intended evaluation objective while maintaining high local resolution along the difficulty axis. We view this as a promising direction for difficulty-aware benchmark design.

References

- [1] Oriol Abril-Pla, Virgile Andreani, Colin Carroll, Larry Dong, Christopher J. Fongesbeck, Maxim Kochurov, Ravin Kumar, Junpeng Lao, Christian C. Luhmann, Osvaldo A. Martin, Michael Osthege, Ricardo Vieira, Thomas Wiecki, and Robert Zinkov. PyMC: A modern and comprehensive probabilistic programming framework in Python. *PeerJ Computer Science*, 9(e1516), 2023.
- [2] Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairish, Areeb Alowisheq, et al. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805, 2024.
- [3] Frank B Baker. *The basics of item response theory*. ERIC, 2001.
- [4] Marthe Ballon, Andres Algaba, Brecht Verbeken, and Vincent Ginis. Estimating problem difficulty without ground truth using large language model comparisons. *arXiv preprint arXiv:2512.14220*, 2025.
- [5] Andrew Bean, Simi Hellsten, Harry Mayne, Jabez Magomere, Ryan Chi, Scott A Hale, Hannah R Kirk, et al. Lingoly: A benchmark of olympiad-level linguistic reasoning puzzles in low resource and extinct languages. *Advances in Neural Information Processing Systems*, 37:26224–26237, 2024.

- [6] Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, et al. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*, 2025.
- [7] Aili Chen, Aonian Li, Baichuan Zhou, Bangwei Gong, Binyang Jiang, Boji Dan, Changqing Yu, Chao Wang, Cheng Ma, Cheng Zhong, et al. The minimax-m2 series: Mini activations unleashing max real-world intelligence. *arXiv preprint arXiv:2605.26494*, 2026.
- [8] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. 2021.
- [9] Yunxiao Chen, Xiaou Li, Jingchen Liu, and Zhiliang Ying. Item response theory—a statistical framework for educational and psychological measurement. *Statistical Science*, 40(2):167–194, 2025.
- [10] Zerui Cheng, Jiashuo Liu, Chunjie Wu, Jianzhu Yao, Pramod Viswanath, Ge Zhang, and Wenhao Huang. Vera: Verified reasoning data augmentation at scale. *arXiv preprint arXiv:2602.13217*, 2026.
- [11] Herbert A David and Haikady N Nagaraja. *Order statistics*. John Wiley & Sons, 2004.
- [12] Mucong Ding, Chenghao Deng, Jocelyn Choo, Zichu Wu, Aakriti Agrawal, Avi Schwarzschild, Tianyi Zhou, Tom Goldstein, John Langford, Anima Anandkumar, et al. Easy2hard-bench: Standardized difficulty labels for profiling llm performance and generalization. *Advances in Neural Information Processing Systems*, 37:44323–44365, 2024.
- [13] Susan E Embretson and Steven P Reise. *Item response theory for psychologists*. Psychology Press, 2013.
- [14] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shiron Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [15] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024.
- [16] Jujie He, Jiakai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.
- [17] David Heineman, Valentin Hofmann, Ian Magnusson, Yuling Gu, Noah A Smith, Hannaneh Hajishirzi, Kyle Lo, and Jesse Dodge. Signal and noise: A framework for reducing uncertainty in language model evaluation. *arXiv preprint arXiv:2508.13144*, 2025.
- [18] 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. *arXiv preprint arXiv:2103.03874*, 2021.
- [19] Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandaraao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. *arXiv preprint arXiv:2504.07086*, 2025.
- [20] 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. *arXiv preprint arXiv:2509.11106*, 2025.
- [21] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

- [22] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.
- [23] 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):6421, 2021.
- [24] Sayash Kapoor, Benedikt Stroebel, Peter Kirgis, Nitya Nadgir, Zachary S Siegel, Boyi Wei, Tianci Xue, Ziru Chen, Felix Chen, Saiteja Utpala, Franck Ndzomga, Dheeraj Oruganty, Sophie Luskin, Kangheng Liu, Botao Yu, Amit Arora, Dongyoon Hahm, Harsh Trivedi, Huan Sun, Juyong Lee, Tengjun Jin, Yifan Mai, Yifei Zhou, Yuxuan Zhu, Rishi Bommasani, Daniel Kang, Dawn Song, Peter Henderson, Yu Su, Percy Liang, and Arvind Narayanan. Holistic agent leaderboard: The missing infrastructure for ai agent evaluation. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.
- [25] Alex Kipnis, Konstantinos Voudouris, Luca M Schulze Buschoff, and Eric Schulz. metabench—a sparse benchmark of reasoning and knowledge in large language models. *arXiv preprint arXiv:2407.12844*, 2024.
- [26] John P Lalor, Hao Wu, and Hong Yu. Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 648–657, 2016.
- [27] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.
- [28] Xinyuan Li, Murong Xu, Wenbiao Tao, Hanlun Zhu, Yike Zhao, Jipeng Zhang, and Yunshi Lan. Ride: Difficulty evolving perturbation with item response theory for mathematical reasoning. *arXiv preprint arXiv:2511.04120*, 2025.
- [29] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.
- [30] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601, 2024.
- [31] Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. ZebraLogic: On the scaling limits of llms for logical reasoning. 2025.
- [32] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.
- [33] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.
- [34] Shu Liu, Shubham Agarwal, Monishwaran Maheswaran, Mert Cemri, Zhifei Li, Qiuyang Mang, Ashwin Naren, Ethan Boneh, Audrey Cheng, Melissa Z Pan, et al. Evox: Meta-evolution for automated discovery. *arXiv preprint arXiv:2602.23413*, 2026.
- [35] Frederic Lord. A theory of test scores. *Psychometric monographs*, 1952.
- [36] Frederic M Lord. *Applications of item response theory to practical testing problems*. Routledge, 2012.
- [37] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.
- [38] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. <https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2>, 2025. Notion Blog.
- [39] Mathematical Association of America. 2023 American Mathematics Competitions 12 (AMC 12). <https://maa.org/student-programs/amc/>, 2023. Accessed 2026-04-25.

- [40] Mathematical Association of America. 2024 American Invitational Mathematics Examination (AIME). <https://maa.org/maa-invitational-competitions/>, 2024. Accessed 2026-04-25.
- [41] Mathematical Association of America. 2025 American Invitational Mathematics Examination (AIME). <https://maa.org/maa-invitational-competitions/>, 2025. Accessed 2026-04-25.
- [42] Mathematical Association of America. 2026 American Invitational Mathematics Examination (AIME). <https://maa.org/maa-invitational-competitions/>, 2026. Accessed 2026-04-25.
- [43] Grégoire Mialon, Clémentine Fourier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *International Conference on Learning Representations*, volume 2024, pages 9025–9049, 2024.
- [44] Seyed Mahed Mousavi, Edoardo Cecchinato, Lucia Hornikova, and Giuseppe Riccardi. Garbage in, reasoning out? why benchmark scores are unreliable and what to do about it. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 1747–1759, 2026.
- [45] Aidar Myrzakhan, Sondos Mahmoud Bsharat, and Zhiqiang Shen. Open-llm-leaderboard: From multi-choice to open-style questions for llms evaluation, benchmark, and arena. *arXiv preprint arXiv:2406.07545*, 2024.
- [46] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity’s last exam. *arXiv preprint arXiv:2501.14249*, 2025.
- [47] Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*, 2024.
- [48] Qi Qian, Chengsong Huang, Jingwen Xu, Changze Lv, Muling Wu, Wenhao Liu, Xiaohua Wang, Zhenghua Wang, Zisu Huang, Muzhao Tian, et al. Benchmark²: Systematic evaluation of llm benchmarks. *arXiv preprint arXiv:2601.03986*, 2026.
- [49] Georg Rasch. *Probabilistic models for some intelligence and attainment tests*. ERIC, 1993.
- [50] David Rein. Can good benchmarks contain mistakes?, 2024.
- [51] 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.
- [52] Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P Lalor, Robin Jia, and Jordan Boyd-Graber. Evaluation examples are not equally informative: How should that change nlp leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503, 2021.
- [53] Asankhaya Sharma. Openevolve: an open-source evolutionary coding agent, 2025. URL <https://github.com/codelion/openevolve>.
- [54] Quan Shi, Michael Tang, Karthik Narasimhan, and Shunyu Yao. Can language models solve olympiad programming? *arXiv preprint arXiv:2404.10952*, 2024.
- [55] Zachary S Siegel, Sayash Kapoor, Nitya Nadgir, Benedikt Stroebel, and Arvind Narayanan. Core-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *arXiv preprint arXiv:2409.11363*, 2024.
- [56] Klaas Sijtsma and Bas T Hemker. Nonparametric polytomous irt models for invariant item ordering, with results for parametric models. *Psychometrika*, 63(2):183–200, 1998.
- [57] Klaas Sijtsma and Brian W Junker. A survey of theory and methods of invariant item ordering. *British Journal of Mathematical and Statistical Psychology*, 49(1):79–105, 1996.
- [58] Charlotte Siska, Katerina Marazopoulou, Melissa Ailem, and James Bono. Examining the robustness of llm evaluation to the distributional assumptions of benchmarks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10406–10421, 2024.
- [59] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

- [60] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.
- [61] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355, 2018.
- [62] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.
- [63] Chunqiu Steven Xia, Yinlin Deng, and Lingming Zhang. Top leaderboard ranking= top coding proficiency, always? evoeval: Evolving coding benchmarks via llm. *arXiv preprint arXiv:2403.19114*, 2024.
- [64] Jiahao Ying, Yixin Cao, Yushi Bai, Qianru Sun, Bo Wang, Wei Tang, Zhaojun Ding, Yizhe Yang, Xuanjing Huang, and Shuicheng Yan. Automating dataset updates towards reliable and timely evaluation of large language models. *Advances in Neural Information Processing Systems*, 37:17106–17132, 2024.
- [65] Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- [66] Yu Yue, Yufeng Yuan, Qiyang Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, Tiantian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.
- [67] Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*, 2023.
- [68] Ziqian Zhang, Xingjian Hu, Yue Huang, Kai Zhang, Ruoxi Chen, Yixin Liu, Qingsong Wen, Kaidi Xu, Xiangliang Zhang, Neil Zhenqiang Gong, et al. Rankllm: Weighted ranking of llms by quantifying question difficulty. *arXiv preprint arXiv:2602.12424*, 2026.
- [69] Hongli Zhou, Hui Huang, Ziqing Zhao, Lvyuan Han, Huicheng Wang, Kehai Chen, Muyun Yang, Wei Bao, Jian Dong, Bing Xu, et al. Lost in benchmarks? rethinking large language model benchmarking with item response theory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35085–35093, 2026.
- [70] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.
- [71] Yubo Zhu, Dongrui Liu, Zecheng Lin, Wei Tong, Sheng Zhong, and Jing Shao. The llm already knows: Estimating llm-perceived question difficulty via hidden representations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1160–1176, 2025.

A Discussion

A.1 Broader Impacts

This work aims to improve the reliability and transparency of benchmark evaluation. By diagnosing distribution mismatch and coverage gaps, our methods can help researchers better understand when benchmark scores provide faithful evidence for model comparison and monitoring training progress. More reliable evaluation may support more transparent model comparison and more informed benchmark design. Since this work focuses on benchmark analysis rather than model deployment or data collection, we do not foresee direct negative societal impacts.

A.2 Extended Related Work

Difficulty estimation. Prior work often derives difficulty labels from human judgment or benchmark metadata. For example, MATH [18] annotates problems with five difficulty levels, GPQA [51] collects post-hoc expert difficulty ratings, and LingOly [5] includes five levels of human difficulty. Such labels are costly to obtain at scale, and they primarily reflect human-perceived difficulty rather than the difficulty exhibited by current LLMs on the questions. Recent work has explored alternative estimation strategies. [71] estimate LLM-perceived question difficulty from hidden representations. [4] ask an LLM to make pairwise difficulty judgments and aggregate them with Bradley–Terry scores. EIP [68] instead jointly infers question difficulty and model competency from observed successes and failures on a model–question bipartite graph. Among these approaches, IRT remains a common foundation for difficulty-aware evaluation because it provides a principled probabilistic framework for jointly modeling item difficulty and model ability directly from response data [9].

B Theoretical Results and Proofs

B.1 Measure Mismatch

We first restate Theorem 2.1 and provide the corresponding proof.

Theorem B.1 (Distribution mismatch controls score distortion and rank stability). *Suppose the performance difference function Δ is L -Lipschitz, i.e.,*

$$|\Delta(z) - \Delta(y)| \leq L|z - y|, \quad \forall z, y \in [0, 1].$$

Then

$$|D_B(\theta', \theta) - D_\mu(\theta', \theta)| \leq L W_1(\nu_B, \mu).$$

Consequently, if

$$|D_\mu(\theta', \theta)| > L W_1(\nu_B, \mu),$$

then

$$\text{sign}(D_B(\theta', \theta)) = \text{sign}(D_\mu(\theta', \theta)),$$

so the benchmark-induced ordering agrees with the target ordering, i.e., no rank flip can occur.

Proof. Fix any coupling $\pi \in \Pi(\nu_B, \mu)$. Since π has marginals ν_B and μ , we can write

$$D_B(\theta', \theta) - D_\mu(\theta', \theta) = \int \Delta(z) d\nu_B(z) - \int \Delta(y) d\mu(y) = \int_{[0,1]^2} (\Delta(z) - \Delta(y)) d\pi(z, y).$$

Taking absolute values and using the Lipschitz property of Δ gives

$$|D_B(\theta', \theta) - D_\mu(\theta', \theta)| \leq \int_{[0,1]^2} |\Delta(z) - \Delta(y)| d\pi(z, y) \leq L \int_{[0,1]^2} |z - y| d\pi(z, y).$$

Since this holds for every coupling $\pi \in \Pi(\nu_B, \mu)$, taking the infimum over π yields

$$|D_B(\theta', \theta) - D_\mu(\theta', \theta)| \leq L W_1(\nu_B, \mu).$$

If $D_\mu(\theta', \theta) > 0$, then

$$D_B(\theta', \theta) \geq D_\mu(\theta', \theta) - |D_B(\theta', \theta) - D_\mu(\theta', \theta)| \geq D_\mu(\theta', \theta) - L W_1(\nu_B, \mu) > 0.$$

Hence $D_B(\theta', \theta) > 0$. The case $D_\mu(\theta', \theta) < 0$ is symmetric. Therefore $D_B(\theta', \theta)$ and $D_\mu(\theta', \theta)$ must have the same sign, so no rank flip can occur. $\square$

Finite-sample evaluation under repeated sampling. In practice, the conditional mean $s_\theta(z)$ is not directly observed. Instead, benchmark scores are estimated from finitely many repeated samples of model outputs. For the i -th benchmark example with difficulty level z_i , and for each independent trial $r \in \{1, \dots, k\}$, let

$$Y_{i,r}^\theta \in [0, 1]$$

denote the observed score of model θ on the r -th sample, with

$$\mathbb{E}[Y_{i,r}^\theta \mid z_i] = s_\theta(z_i).$$

Define the k -sample benchmark score as

$$\hat{S}_{B,k}(\theta) := \frac{1}{nk} \sum_{i=1}^n \sum_{r=1}^k Y_{i,r}^\theta,$$

and the corresponding observed performance difference as

$$\hat{D}_{B,k}(\theta', \theta) := \hat{S}_{B,k}(\theta') - \hat{S}_{B,k}(\theta).$$

Theorem B.2 (Finite-sample score distortion and ordering stability under repeated sampling). *Suppose the performance difference function Δ is L -Lipschitz. Assume that, conditional on the difficulty levels $z_1, \dots, z_n$, the collection of random variables*

$$\left\{ Y_{i,r}^\theta, Y_{i,r}^{\theta'} \right\}_{i=1, \dots, n; r=1, \dots, k}$$

is independent, and that each variable takes values in $[0, 1]$. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$|\hat{D}_{B,k}(\theta', \theta) - D_\mu(\theta', \theta)| \leq L W_1(\nu_B, \mu) + \sqrt{\frac{2 \log(2/\delta)}{nk}}.$$

Consequently, if

$$|D_\mu(\theta', \theta)| > L W_1(\nu_B, \mu) + \sqrt{\frac{2 \log(2/\delta)}{nk}},$$

then with probability at least $1 - \delta$,

$$\text{sign}(\hat{D}_{B,k}(\theta', \theta)) = \text{sign}(D_\mu(\theta', \theta)),$$

and hence no observed rank flip occurs.

Proof. We first decompose the error as

$$\hat{D}_{B,k}(\theta', \theta) - D_\mu(\theta', \theta) = \left(\hat{D}_{B,k}(\theta', \theta) - D_B(\theta', \theta) \right) + \left(D_B(\theta', \theta) - D_\mu(\theta', \theta) \right).$$

Therefore,

$$|\hat{D}_{B,k}(\theta', \theta) - D_\mu(\theta', \theta)| \leq |\hat{D}_{B,k}(\theta', \theta) - D_B(\theta', \theta)| + |D_B(\theta', \theta) - D_\mu(\theta', \theta)|.$$

The second term is deterministic and, by Theorem 2.1, satisfies

$$|D_B(\theta', \theta) - D_\mu(\theta', \theta)| \leq L W_1(\nu_B, \mu).$$

It remains to bound the first term. Define

$$Z_{i,r} := Y_{i,r}^{\theta'} - Y_{i,r}^\theta.$$

Since $Y_{i,r}^{\theta'}, Y_{i,r}^\theta \in [0, 1]$, we have

$$Z_{i,r} \in [-1, 1].$$

Moreover, conditional on z_i ,

$$\mathbb{E}[Z_{i,r} \mid z_i] = \mathbb{E}[Y_{i,r}^{\theta'} \mid z_i] - \mathbb{E}[Y_{i,r}^\theta \mid z_i] = s_{\theta'}(z_i) - s_\theta(z_i) = \Delta(z_i).$$

Hence

$$\hat{D}_{B,k}(\theta', \theta) - D_B(\theta', \theta) = \frac{1}{nk} \sum_{i=1}^n \sum_{r=1}^k (Z_{i,r} - \Delta(z_i)).$$

Conditional on $z_1, \dots, z_n$, the variables $\{Z_{i,r}\}$ are independent and each lies in $[-1, 1]$. Applying Hoeffding's inequality conditionally yields, for any $t > 0$,

$$\mathbb{P}\left(|\hat{D}_{B,k}(\theta', \theta) - D_B(\theta', \theta)| \geq t \mid z_1, \dots, z_n\right) \leq 2 \exp\left(-\frac{nk t^2}{2}\right).$$

Setting

$$t = \sqrt{\frac{2 \log(2/\delta)}{nk}}$$

gives, with probability at least $1 - \delta$,

$$|\widehat{D}_{B,k}(\theta', \theta) - D_B(\theta', \theta)| \leq \sqrt{\frac{2 \log(2/\delta)}{nk}}.$$

Combining this bound with the deterministic bound above proves

$$|\widehat{D}_{B,k}(\theta', \theta) - D_\mu(\theta', \theta)| \leq L W_1(\nu_B, \mu) + \sqrt{\frac{2 \log(2/\delta)}{nk}}$$

with probability at least $1 - \delta$.

For the second claim, if

$$|D_\mu(\theta', \theta)| > L W_1(\nu_B, \mu) + \sqrt{\frac{2 \log(2/\delta)}{nk}},$$

then the above inequality implies that $\widehat{D}_{B,k}(\theta', \theta)$ cannot cross zero, so it must have the same sign as $D_\mu(\theta', \theta)$. Therefore no observed rank flip occurs. $\square$

Uniform distribution as a principled target distribution without prior information. The following result formalizes the uniform distribution as a principled default target when no prior knowledge is available about where performance differences arise along the difficulty axis. We consider an agnostic setting in which the signed performance difference Δ has a fixed average level ρ over $[0, 1]$, but its location is otherwise unrestricted. For a candidate target distribution μ , the quantity $V_\rho(\mu)$ measures the worst-case amount of difference retained under μ . The theorem below shows that the uniform distribution is the unique choice that preserves the full average difference ρ in the worst case.

Theorem B.3 (Uniform target as the unique maximin choice under complete uncertainty). *Let*

$$\mathcal{F}_\rho := \left\{ \Delta : [0, 1] \rightarrow [-1, 1] \text{ measurable} : \int_0^1 \Delta(z) dz = \rho \right\}, \quad V_\rho(\mu) := \inf_{\Delta \in \mathcal{F}_\rho} \int \Delta(z) d\mu(z),$$

where $\rho \in (-1, 1)$ and $\mu \in \mathcal{P}([0, 1])$. Then

$$\sup_{\mu \in \mathcal{P}([0, 1])} V_\rho(\mu) = \rho$$

and the unique maximizer is the uniform distribution $\lambda = \text{Unif}[0, 1]$.

Proof. Set

$$\beta := \frac{1 + \rho}{2} \in (0, 1).$$

For any measurable function $\Delta : [0, 1] \rightarrow [-1, 1]$, define

$$G(z) := \frac{\Delta(z) + 1}{2}.$$

Then $G : [0, 1] \rightarrow [0, 1]$ is measurable, and

$$\int_0^1 G(z) dz = \frac{1}{2} \int_0^1 \Delta(z) dz + \frac{1}{2}.$$

Therefore, $\Delta \in \mathcal{F}_\rho$ if and only if

$$G : [0, 1] \rightarrow [0, 1] \quad \text{and} \quad \int_0^1 G(z) dz = \beta.$$

Thus, for any $\mu \in \mathcal{P}([0, 1])$,

$$V_\rho(\mu) = \inf_{\Delta \in \mathcal{F}_\rho} \int \Delta(z) d\mu(z) = \inf_{G : [0, 1] \rightarrow [0, 1], \int_0^1 G(z) dz = \beta} \int (2G(z) - 1) d\mu(z).$$

Since μ is a probability measure,

$$V_\rho(\mu) = 2 \inf_{G : [0, 1] \rightarrow [0, 1], \int_0^1 G(z) dz = \beta} \int G(z) d\mu(z) - 1.$$

Define

$$U_\beta(\mu) := \inf_{G : [0, 1] \rightarrow [0, 1], \int_0^1 G(z) dz = \beta} \int G(z) d\mu(z).$$

Then

$$V_\rho(\mu) = 2U_\beta(\mu) - 1.$$

We now characterize $\sup_{\mu \in \mathcal{P}([0,1])} U_\beta(\mu)$.

First, if $\mu = \lambda = \text{Unif}[0, 1]$, then for every admissible G ,

$$\int G(z) d\mu(z) = \int_0^1 G(z) dz = \beta,$$

so

$$U_\beta(\lambda) = \beta.$$

Hence

$$\sup_{\mu \in \mathcal{P}([0,1])} U_\beta(\mu) \geq \beta.$$

Next fix any $\mu \in \mathcal{P}([0, 1])$. By the Lebesgue decomposition theorem, we may write

$$\mu = f(z) dz + \mu_s,$$

where $f \geq 0$ is integrable and $\mu_s \perp dz$. Let

$$m := \int_0^1 f(z) dz = \mu_{\text{ac}}([0, 1]) \leq 1.$$

Since μ_s is singular with respect to Lebesgue measure, there exists a measurable set $N \subseteq [0, 1]$ such that

$$|N| = 0 \quad \text{and} \quad \mu_s([0, 1] \setminus N) = 0.$$

Thus, for any admissible G that vanishes on N , we have

$$\int G(z) d\mu_s(z) = 0.$$

Therefore,

$$U_\beta(\mu) \leq \inf_{\substack{0 \leq G \leq 1, \\ G=0 \text{ on } N, \\ \int_0^1 G(z) dz = \beta}} \int_0^1 G(z) f(z) dz.$$

Since N has Lebesgue measure zero, the additional constraint $G = 0$ on N does not affect the infimum, so

$$U_\beta(\mu) \leq \inf_{0 \leq G \leq 1, \int_0^1 G(z) dz = \beta} \int_0^1 G(z) f(z) dz.$$

By the bathtub principle, the right-hand side equals

$$\int_0^\beta f^\uparrow(u) du,$$

where $f^\uparrow$ is the nondecreasing rearrangement of f . Since $f^\uparrow$ is nondecreasing and $\beta \in (0, 1)$,

$$\frac{1}{\beta} \int_0^\beta f^\uparrow(u) du \leq \int_0^1 f^\uparrow(u) du = \int_0^1 f(z) dz \leq 1.$$

Hence

$$U_\beta(\mu) \leq \beta.$$

Since this holds for every $\mu \in \mathcal{P}([0, 1])$, we conclude that

$$\sup_{\mu \in \mathcal{P}([0,1])} U_\beta(\mu) = \beta.$$

Consequently,

$$\sup_{\mu \in \mathcal{P}([0,1])} V_\rho(\mu) = 2 \sup_{\mu \in \mathcal{P}([0,1])} U_\beta(\mu) - 1 = 2\beta - 1 = \rho.$$

It remains to prove uniqueness. Suppose $V_\rho(\mu) = \rho$. Then necessarily

$$U_\beta(\mu) = \beta.$$

From the argument above, this implies

$$\int_0^\beta f^\uparrow(u) du = \beta \quad \text{and} \quad \int_0^1 f(z) dz = 1.$$

The second condition implies $\mu_s = 0$, so μ is absolutely continuous with density f . Moreover,

$$\frac{1}{\beta} \int_0^\beta f^\uparrow(u) du = 1$$

equals the average of $f^\uparrow$ over the whole interval $[0, 1]$, which is also 1. Since $f^\uparrow$ is nondecreasing, this is possible only if

$$f^\uparrow(u) = 1 \quad \text{for a.e. } u \in [0, 1].$$

Hence $f(z) = 1$ for a.e. $z \in [0, 1]$, and therefore $\mu = \lambda$. This proves uniqueness. $\square$

B.2 Coverage Gaps

To analyze whether local improvements can be detected from benchmark evaluations, we formulate the problem as a hypothesis test comparing two performance curves before and after improvement. We consider a generic pair of pre- and post-improvement curves

$$s^-, s^+ : [0, 1] \rightarrow [0, 1].$$

Here, $s^-(z)$ and $s^+(z)$ represent the expected model performance (e.g., pass@1) at difficulty level z before and after improvement, respectively. Their difference is

$$\Delta(z) := s^+(z) - s^-(z).$$

For the i -th benchmark example and each trial $r \in \{1, \dots, k\}$, let

$$Y_{i,r}^- \sim \text{Bernoulli}(s^-(z_i)), \quad Y_{i,r}^+ \sim \text{Bernoulli}(s^+(z_i)).$$

We assume that, conditional on the difficulty levels and the performance curves, all variables

$$\{Y_{i,r}^-, Y_{i,r}^+\}_{i=1, \dots, n; r=1, \dots, k}$$

are mutually independent. Define

$$\widehat{S}_{B,k}(s^-) := \frac{1}{nk} \sum_{i=1}^n \sum_{r=1}^k Y_{i,r}^-, \quad \widehat{S}_{B,k}(s^+) := \frac{1}{nk} \sum_{i=1}^n \sum_{r=1}^k Y_{i,r}^+,$$

and

$$\widehat{D}_{B,k}(s^+, s^-) := \widehat{S}_{B,k}(s^+) - \widehat{S}_{B,k}(s^-).$$

Fix $w \in (0, 1]$ and $a \in (0, 1]$. We define the localized alternative class

$$\mathcal{H}_1(w, a) := \left\{ (s^-, s^+) : \exists \text{ interval } I \subset [0, 1], |I| = w, \text{ such that } s^+(z) - s^-(z) \geq a \ \forall z \in I \right\}.$$

Here, the parameters (w, a) characterize the improvement pattern: w represents the width of the difficulty region where the improvement occurs, and a represents the minimum magnitude of improvement within that region.

The null class is

$$\mathcal{H}_0 := \{(s^-, s^+) : s^- = s^+\}.$$

Thus we consider the testing problem

$$H_0 : (s^-, s^+) \in \mathcal{H}_0 \quad \text{versus} \quad H_1 : (s^-, s^+) \in \mathcal{H}_1(w, a).$$

This formulation captures the question of whether a benchmark can reliably detect localized meaningful improvements of width w and magnitude at least a .

A (possibly randomized) test ϕ has level at most $\alpha \in (0, 1)$ if

$$\sup_{(s^-, s^+) \in \mathcal{H}_0} \mathbb{E}_{s^-, s^+}[\phi] \leq \alpha.$$

Its worst-case power over $\mathcal{H}_1(w, a)$ is

$$\inf_{(s^-, s^+) \in \mathcal{H}_1(w, a)} \mathbb{E}_{s^-, s^+}[\phi].$$

We define the minimax worst-case power by

$$\Pi_\alpha^*(B; w, a, k) := \sup_{\phi: \sup_{(s^-, s^+) \in \mathcal{H}_0} \mathbb{E}_{s^-, s^+}[\phi] \leq \alpha} \inf_{(s^-, s^+) \in \mathcal{H}_1(w, a)} \mathbb{E}_{s^-, s^+}[\phi].$$

Theorem B.4 (Maximum gap as an exact asymptotic minimax detection boundary). *Fix $\alpha \in (0, 1)$ and $a \in (0, 1]$.*

1. *For every $k \in \mathbb{N}$, if $w \leq G(B)$, then*

$$\Pi_\alpha^*(B; w, a, k) = \alpha.$$

2. *If $w > G(B)$, then*

$$\lim_{k \rightarrow \infty} \Pi_\alpha^*(B; w, a, k) = 1.$$

Therefore, $G(B)$ is the exact asymptotic minimax detection boundary for interval-localized differences under benchmark B .

Proof. We prove the two parts separately.

Part 1: if $w \leq G(B)$, then $\Pi_\alpha^*(B; w, a, k) = \alpha$.

By definition of the maximum gap, there exists an index $j \in \{0, \dots, n\}$ such that

$$z_{(j+1)} - z_{(j)} \geq w.$$

Let

$$J := (z_{(j)}, z_{(j+1)}).$$

Since

$$|J| = z_{(j+1)} - z_{(j)} \geq w,$$

we may choose an interval

$$I \subseteq J \quad \text{such that} \quad |I| = w.$$

Choose any $c \in [0, 1 - a]$, and define

$$s^{-,*}(z) \equiv c, \quad s^{+,*}(z) := c + a \mathbf{1}\{z \in I\}.$$

Since $c \in [0, 1 - a]$ and $a \in (0, 1]$, both $s^{-,*}$ and $s^{+,*}$ take values in $[0, 1]$. Moreover,

$$s^{+,*}(z) - s^{-,*}(z) = a \quad \forall z \in I,$$

so

$$(s^{-,*}, s^{+,*}) \in \mathcal{H}_1(w, a).$$

Because the interval I contains no benchmark point, for every $i = 1, \dots, n$ we have

$$s^{-,*}(z_i) = c, \quad s^{+,*}(z_i) = c.$$

Therefore, under $(s^{-,*}, s^{+,*})$, the benchmark observations have exactly the same joint distribution as under the null pair

$$(\bar{s}, \bar{s}) \in \mathcal{H}_0 \quad \text{with} \quad \bar{s}(z) \equiv c.$$

Indeed, at every observed benchmark point z_i , both the pre- and post-improvement Bernoulli success probabilities are equal to c .

Now let ϕ be any test of level at most α . Then

$$\mathbb{E}_{s^{-,*}, s^{+,*}}[\phi] = \mathbb{E}_{\bar{s}, \bar{s}}[\phi] \leq \sup_{(s^-, s^+) \in \mathcal{H}_0} \mathbb{E}_{s^-, s^+}[\phi] \leq \alpha.$$

Hence

$$\inf_{(s^-, s^+) \in \mathcal{H}_1(w, a)} \mathbb{E}_{s^-, s^+}[\phi] \leq \mathbb{E}_{s^{-,*}, s^{+,*}}[\phi] \leq \alpha.$$

Taking the supremum over all level- α tests yields

$$\Pi_\alpha^*(B; w, a, k) \leq \alpha.$$

For the reverse inequality, consider the trivial randomized test that rejects with probability α independently of the data. It has level exactly α , and for every alternative its power is also exactly α . Therefore

$$\Pi_\alpha^*(B; w, a, k) \geq \alpha.$$

Combining the two inequalities, we conclude

$$\Pi_\alpha^*(B; w, a, k) = \alpha.$$

Part 2: if $w > G(B)$, then $\lim_{k \rightarrow \infty} \Pi_\alpha^*(B; w, a, k) = 1$.

We construct an explicit level- α test whose worst-case power tends to 1 as $k \rightarrow \infty$.

For each benchmark point z_i , define the empirical difference

$$\hat{\Delta}_{i,k} := \frac{1}{k} \sum_{r=1}^k (Y_{i,r}^+ - Y_{i,r}^-).$$

Consider the max test

$$\phi_{\max} := \mathbf{1} \left\{ \max_{1 \leq i \leq n} \hat{\Delta}_{i,k} > t_{k,\alpha} \right\}, \quad t_{k,\alpha} := \sqrt{\frac{2 \log(n/\alpha)}{k}}.$$

We first verify that $\phi_{\max}$ has level at most α . Under any null pair $(s, s) \in \mathcal{H}_0$, for each fixed i and r ,

$$Y_{i,r}^+ - Y_{i,r}^- \in [-1, 1], \quad \mathbb{E}_{s,s}[Y_{i,r}^+ - Y_{i,r}^-] = 0,$$

and these random variables are independent across r . Hence, by Hoeffding's inequality,

$$\mathbb{P}_{s,s}(\widehat{\Delta}_{i,k} > t) \leq \exp\left(-\frac{kt^2}{2}\right).$$

Applying the union bound over $i = 1, \dots, n$ gives

$$\mathbb{P}_{s,s}(\phi_{\max} = 1) = \mathbb{P}_{s,s}\left(\max_{1 \leq i \leq n} \widehat{\Delta}_{i,k} > t_{k,\alpha}\right) \leq \sum_{i=1}^n \mathbb{P}_{s,s}(\widehat{\Delta}_{i,k} > t_{k,\alpha}) \leq n \exp\left(-\frac{kt_{k,\alpha}^2}{2}\right) = \alpha.$$

Since this bound is uniform over $(s, s) \in \mathcal{H}_0$, the test $\phi_{\max}$ has level at most α .

Next fix any alternative pair $(s^-, s^+) \in \mathcal{H}_1(w, a)$. By definition, there exists an interval $I \subset [0, 1]$ with $|I| = w$ such that

$$s^+(z) - s^-(z) \geq a, \quad \forall z \in I.$$

Because $w > G(B)$, no interval of length w can be disjoint from all benchmark points. Therefore, there exists at least one benchmark index i^* such that

$$z_{i^*} \in I.$$

For this index we have

$$\mathbb{E}_{s^-, s^+}[\widehat{\Delta}_{i^*,k}] = s^+(z_{i^*}) - s^-(z_{i^*}) \geq a.$$

Again by Hoeffding's inequality, for any $t \in \mathbb{R}$,

$$\mathbb{P}_{s^-, s^+}(\widehat{\Delta}_{i^*,k} \leq t) \leq \exp\left(-\frac{k}{2}(a-t)_+^2\right),$$

where $(u)_+ := \max\{u, 0\}$. Setting $t = t_{k,\alpha}$ yields

$$\mathbb{P}_{s^-, s^+}(\widehat{\Delta}_{i^*,k} \leq t_{k,\alpha}) \leq \exp\left(-\frac{k}{2}(a-t_{k,\alpha})_+^2\right).$$

Since the event

$$\{\widehat{\Delta}_{i^*,k} > t_{k,\alpha}\}$$

implies

$$\left\{\max_{1 \leq i \leq n} \widehat{\Delta}_{i,k} > t_{k,\alpha}\right\},$$

we obtain

$$\mathbb{P}_{s^-, s^+}(\phi_{\max} = 1) \geq \mathbb{P}_{s^-, s^+}(\widehat{\Delta}_{i^*,k} > t_{k,\alpha}) \geq 1 - \exp\left(-\frac{k}{2}(a-t_{k,\alpha})_+^2\right).$$

This lower bound is uniform over all $(s^-, s^+) \in \mathcal{H}_1(w, a)$, so

$$\inf_{(s^-, s^+) \in \mathcal{H}_1(w, a)} \mathbb{E}_{s^-, s^+}[\phi_{\max}] \geq 1 - \exp\left(-\frac{k}{2}(a-t_{k,\alpha})_+^2\right).$$

Since the minimax power is at least the worst-case power of any specific level- α test,

$$\Pi_\alpha^*(B; w, a, k) \geq 1 - \exp\left(-\frac{k}{2}(a-t_{k,\alpha})_+^2\right).$$

Finally, because

$$t_{k,\alpha} = \sqrt{\frac{2 \log(n/\alpha)}{k}} \rightarrow 0 \quad \text{as } k \rightarrow \infty,$$

the right-hand side converges to 1. Since $\Pi_\alpha^*(B; w, a, k) \leq 1$ by definition of power, the squeeze theorem implies

$$\lim_{k \rightarrow \infty} \Pi_\alpha^*(B; w, a, k) = 1.$$

This completes the proof. $\square$

C Difficulty Estimation Details and Additional Results

C.1 Benchmark Details

Tab. 3 summarizes the benchmarks used in our analysis, including their domains, the number of examples, and the number of rollouts per item during evaluation.

Table 3: **Summary of the benchmarks used in our analysis.** For compactness, we abbreviate SWE-bench Verified (Mini) as SWE-bench-Mini, and BIG-Bench Hard-Logical Deduction- $\{3,5,7\}$ objects as BBH- $\{3,5,7\}$ objects throughout the paper.

Benchmark	Domain	# Examples	# Rollouts / Item
MATH-500 [30]	Math	500	8
Minerva-Math [27]	Math	272	8
OlympiadBench [15]	Math	675	8
Gaokao2023en [67]	Math	385	8
AMC23 [39]	Math	40	32
AIME24 [40]	Math	30	32
AIME25 [41]	Math	30	32
AIME26 [42]	Math	30	32
HumanEval [8]	Coding	164	4
MBPP+ [32]	Coding	378	4
GAIA [43]	Agentic Tasks	165	1
CORE-Bench Hard [55]	Agentic Tasks	45	1
SWE-bench Verified (Mini) [22]	Agentic Tasks	50	1
USACO [54]	Agentic Tasks	307	1
MMLU-Pro-Biology [62]	Science	717	4
MMLU-Pro-Chemistry [62]	Science	1132	4
MMLU-Pro-Physics [62]	Science	1299	4
GPQA-Diamond [51]	Science	198	4
ScienceQA [37]	Science	4241	4
IFEval [70]	Instruction Following	541	4
BIG-Bench Hard-Logical Deduction-3 objects [59]	Logic	250	4
BIG-Bench Hard-Logical Deduction-5 objects [59]	Logic	250	4
BIG-Bench Hard-Logical Deduction-7 objects [59]	Logic	250	4
ZebraLogic [31]	Logic	1000	4
MedQA [23]	Medicine	1273	4

C.2 Model Selection Details

For non-agentic tasks, we select models based on the overall leaderboard of Text Arena, whose evaluation suite covers a diverse set of domains including Expert, Hard Prompts, Coding, Math, Creative Writing, Instruction Following, and Longer Query. The overall score therefore provides a reasonable proxy for model capability across a broad range of tasks. To obtain models with diverse capability levels, we partition the leaderboard into 10 equal-width tiers according to the overall score. We then randomly select one model from each tier and additionally include one model from both the top and bottom of the leaderboard. This results in a total of 12 models spanning a wide spectrum of capabilities. The detailed model information is summarized in Tab. 4.

Table 4: **Summary of selected models for non-agentic tasks.**

Model	Size	Provider	Temperature	Top- p
Qwen3.5-397B-A17B	397B	Alibaba Qwen	0.6	0.95
GLM-4.6	357B	Zhipu AI	1.0	0.95
GLM-4.5-Air	110B	Zhipu AI	0.6	0.95
Qwen3-Next-80B-A3B-Instruct	80B	Alibaba Qwen	0.7	0.8
QwQ-32B	32B	Alibaba Qwen	0.6	0.95
Olmo-3.1-32B-Think	32B	Allen AI	0.6	0.95
gpt-oss-20b	20B	OpenAI	1.0	1.0
phi-4	14B	Microsoft	0.8	0.95
Llama-3.1-8B-Instruct	8B	Meta-Llama	0.6	0.9
vicuna-7b-v1.5	7B	LMSYS	0.9	0.6
Llama-3.2-3B-Instruct	3B	Meta-Llama	0.6	0.9
SmolLM2-1.7B-Instruct	1.7B	HuggingFaceTB	0.7	0.8

For agentic tasks, we use models included in the HAL evaluation platform. These include several frontier proprietary models from major providers, such as Claude Sonnet 4.5, Claude Opus 4.1, GPT-5, and Gemini 2.0 Flash.

C.3 IRT Fitting and Diagnostics

To obtain reliable difficulty estimates, we conduct several diagnostic checks on both the response matrices and the IRT fitting procedure.

Response matrix diagnostics. We examine each response matrix using four diagnostics. The first two, *model information* and *item information*, measure whether the matrix contains enough discriminative signal along the model and item dimensions [3]. The latter two, *model inversion rate* and *item inversion rate*, measure how well the empirical response matrix agrees with the monotone structure implied by 1PL-IRT [56, 57].

Formally, for each item–model pair (i, j) , we aggregate the observed responses into the empirical success rate of model j on item i , defined as $p_{ij} = n_{ij}^{\text{correct}} / n_{ij}^{\text{trials}}$. We then aggregate these success rates along the model and item dimensions:

$$p_j = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} p_{ij}, \quad p_i = \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} p_{ij},$$

where $\mathcal{I}$ and $\mathcal{J}$ denote the sets of items and models, respectively. We define model information as the average of $4p_j(1 - p_j)$ over models, and item information as the average of $4p_i(1 - p_i)$ over items. The intuition is that if many models solve nearly all items or almost none of them, then their aggregate success rates are saturated and contribute limited information for distinguishing model abilities. The same holds when many items are solved by nearly all models or by almost none. In contrast, aggregate success rates near 0.5 are more informative because they correspond to models or items that are neither saturated nor degenerate. The factor 4 rescales this quantity to $[0, 1]$.

To evaluate whether the response matrix is consistent with the ordering implied by 1PL-IRT, we first rank models by overall success rate and items by overall solve rate. We then compute two inversion rates. The model inversion rate is the fraction of ordered model pairs, evaluated item by item, for which a stronger model performs worse than a weaker one. The item inversion rate is the fraction of ordered item pairs, evaluated model by model, for which an easier item is solved less often than a harder one. Lower inversion rates indicate better agreement with the latent ordering assumed by IRT.

In our experiments, we use model information ≥ 0.4 , item information ≥ 0.4 , model inversion rate ≤ 0.3 , and item inversion rate ≤ 0.3 as basic thresholds. Passing these checks provides evidence that the matrix contains sufficient signal and is reasonably consistent with the ordering assumptions of IRT.

IRT fitting. We fit the 1PL-IRT model using Bayesian inference. For multiple-choice benchmarks, we additionally include a guessing parameter:

$$\Pr(Y_{ij} = 1) = g_i + (1 - g_i)\sigma(\theta_j - b_i),$$

where g_i denotes the chance-level probability of answering item i correctly by guessing.

Model abilities θ_j and item difficulties b_i are assigned independent Gaussian priors,

$$\theta_j \sim \mathcal{N}(0, 1), \quad b_i \sim \mathcal{N}(0, 1).$$

Because each model–item pair is evaluated multiple times, we use a binomial likelihood to model the number of correct responses. Posterior inference is performed using MCMC with the No-U-Turn Sampler (NUTS) implemented in PyMC [1]. We run four chains with 1,000 warmup iterations and 1,000 posterior samples per chain. We set the target acceptance rate to 0.95.

We also explored alternative prior structures to assess robustness. In particular, we considered a hierarchical prior for model abilities,

$$\theta_j \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2), \quad \mu_\theta \sim \mathcal{N}(0, 1), \quad \sigma_\theta \sim \text{HalfNormal}(1),$$

while keeping item difficulties under the same Gaussian prior as above. We additionally tested a hierarchical prior for item difficulties,

$$b_i \sim \mathcal{N}(\mu_b, \sigma_b^2), \quad \mu_b \sim \mathcal{N}(0, 1), \quad \sigma_b \sim \text{HalfNormal}(1),$$

with model abilities using the standard Gaussian prior. Across these variants, the estimated item difficulties and model abilities were highly consistent. Therefore, we report results using the simplest specification with independent $\mathcal{N}(0, 1)$ priors.

IRT diagnostics. To assess convergence and sampling quality, we monitor 3 standard MCMC diagnostics: the maximum Gelman–Rubin statistic $\hat{R}$ across parameters, the minimum bulk effective sample size (ESS), and the number of divergent transitions. The $\hat{R}$ statistic measures agreement across chains, with values close to 1 indicating good convergence. Bulk ESS quantifies the effective number of independent posterior samples in the main mass of the distribution, with larger values indicating more reliable estimation. Divergent transitions indicate potential numerical or geometric pathologies during NUTS sampling, and zero divergent transitions are preferred.

Diagnostic results. Tabs. 5 and 6 summarize the diagnostic results for all benchmarks. All benchmarks pass the response matrix diagnostics and the MCMC convergence diagnostics described above.

Table 5: **Response matrix diagnostic results.** $\uparrow$ indicates that larger values are preferred and $\downarrow$ indicates that smaller values are preferred. We use model information ≥ 0.4 , item information ≥ 0.4 , model inversion rate ≤ 0.3 , and item inversion rate ≤ 0.3 as basic thresholds for response matrix diagnostics. All benchmarks pass these checks.

Benchmark	Model Information $\uparrow$	Item Information $\uparrow$	Model Inversion Rate $\downarrow$	Item Inversion Rate $\downarrow$
AMC23	0.431	0.895	0.062	0.177
AIME24	0.514	0.824	0.040	0.101
AIME25	0.515	0.806	0.057	0.118
AIME26	0.465	0.831	0.046	0.089
Gaokao2023en	0.661	0.681	0.101	0.112
HumanEval	0.546	0.675	0.165	0.205
MATH-500	0.463	0.730	0.063	0.125
MBPP+	0.665	0.636	0.226	0.235
Minerva-Math	0.724	0.554	0.174	0.107
OlympiadBench	0.715	0.663	0.106	0.093
BBH-3 objects	0.413	0.617	0.069	0.231
BBH-5 objects	0.421	0.735	0.076	0.292
BBH-7 objects	0.451	0.790	0.080	0.285
GPQA-Diamond	0.843	0.852	0.249	0.251
IFEval	0.663	0.573	0.217	0.162
MedQA	0.795	0.777	0.264	0.262
MMLU-Pro-Biology	0.614	0.613	0.172	0.160
MMLU-Pro-Chemistry	0.676	0.787	0.162	0.186
MMLU-Pro-Physics	0.687	0.787	0.176	0.195
ScienceQA	0.659	0.434	0.184	0.102
ZebraLogic	0.581	0.796	0.048	0.060
GAIA	0.879	0.739	0.251	0.178
CORE-Bench Hard	0.651	0.578	0.192	0.135
SWE-bench-Mini	0.709	0.674	0.112	0.103
USACO	0.908	0.472	0.205	0.059

Table 6: **MCMC diagnostic results for IRT fitting.** $\uparrow$ indicates that larger values are preferred and $\downarrow$ indicates that smaller values are preferred.

Benchmark	Max $\hat{R}$ $\downarrow$	Min Bulk ESS $\uparrow$	# Divergent Transitions $\downarrow$
AMC23	1.01	334	0
AIME24	1.02	272	0
AIME25	1.01	246	0
AIME26	1.00	385	0
Gaokao2023en	1.01	569	0
HumanEval	1.01	826	0
MATH-500	1.01	1296	0
MBPP+	1.01	1167	0
Minerva-Math	1.01	487	0
OlympiadBench	1.01	1246	0
BBH-3 objects	1.00	1996	0
BBH-5 objects	1.01	1450	0
BBH-7 objects	1.00	1218	0
GPQA-Diamond	1.00	1437	0
IFEval	1.01	2120	0
MedQA	1.01	1733	0
MMLU-Pro-Biology	1.01	2511	0
MMLU-Pro-Chemistry	1.01	2014	0
MMLU-Pro-Physics	1.01	2002	0
ScienceQA	1.01	1757	0
ZebraLogic	1.01	1960	0
GAIA	1.00	1861	0
CORE-Bench Hard	1.01	3272	0
SWE-bench-Mini	1.00	4419	0
USACO	1.01	2102	0

Leave-one-out robustness analysis. For each non-agentic benchmark, we assess the robustness of IRT-based difficulty estimation by performing a leave-one-out analysis over the selected model set. Specifically, given the 12 selected models, we refit IRT 12 times, each time removing one model and re-estimating item difficulties from the remaining 11 models. We then compare the resulting difficulty estimates with the original estimates obtained from all 12 models using Pearson and Spearman correlations. For each benchmark, we report

Table 7: **Leave-one-out robustness of IRT difficulty estimation.** For each benchmark, we fit IRT using the full set of 12 models, then refit IRT after removing one model at a time and compare the resulting item difficulties with those from the full fit. We report the mean and minimum Pearson and Spearman correlations across the 12 leave-one-out refits. The consistently high correlations indicate that our selected 12 models are sufficient for stable item difficulty estimation.

Benchmark	Pearson Mean	Pearson Min	Spearman Mean	Spearman Min
AMC23	0.9962	0.9820	0.9921	0.9484
AIME24	0.9985	0.9940	0.9977	0.9920
AIME25	0.9978	0.9904	0.9941	0.9809
AIME26	0.9979	0.9937	0.9972	0.9888
Gaokao2023en	0.9974	0.9953	0.9957	0.9905
HumanEval	0.9902	0.9771	0.9871	0.9735
MATH-500	0.9952	0.9896	0.9935	0.9844
MBPP+	0.9902	0.9751	0.9840	0.9599
Minerva-Math	0.9974	0.9951	0.9896	0.9871
OlympiadBench	0.9980	0.9953	0.9973	0.9946
BBH-3 objects	0.9766	0.9137	0.9764	0.9089
BBH-5 objects	0.9723	0.9207	0.9704	0.9162
BBH-7 objects	0.9764	0.9360	0.9745	0.9248
GPQA-Diamond	0.9941	0.9893	0.9937	0.9852
IFEval	0.9945	0.9853	0.9910	0.9806
MedQA	0.9916	0.9875	0.9861	0.9722
MMLU-Pro-Biology	0.9951	0.9894	0.9923	0.9792
MMLU-Pro-Chemistry	0.9955	0.9914	0.9936	0.9860
MMLU-Pro-Physics	0.9952	0.9907	0.9934	0.9839
ScienceQA	0.9965	0.9893	0.9869	0.9330
ZebraLogic	0.9974	0.9948	0.9966	0.9909
Average	0.9926	0.9798	0.9897	0.9696

both the average and the minimum correlation across the 12 leave-one-out refits. As shown in Tab. 7, the average Pearson and Spearman correlations are at least 0.9723 and 0.9704, respectively, across all benchmarks. Even the worst-case leave-one-out correlation remains high, with the minimum Pearson and Spearman correlations being 0.9137 and 0.9089, respectively. These results indicate that the estimated item difficulties are stable to the exclusion of any single model, supporting the adequacy of our 12-model selection for stable difficulty estimation.

C.4 Metric Calculation Details

For a benchmark with n items, we compare its normalized difficulty points to a same-size empirical uniform reference on $[0, 1]$. Let $z_{(1)} \leq \dots \leq z_{(n)}$ denote the sorted normalized difficulties. We define the uniform reference points as the midpoints of n equal-width bins:

$$u_i = \frac{i - \frac{1}{2}}{n}, \quad i = 1, \dots, n.$$

The DDM is then computed as

$$\text{DDM}(B) = \frac{1}{n} \sum_{i=1}^n |z_{(i)} - u_i|.$$

This is the one-dimensional 1-Wasserstein distance between the benchmark empirical measure and the same-size empirical uniform reference

$$\lambda_n = \frac{1}{n} \sum_{i=1}^n \delta_{u_i}.$$

For the gap-based metric, let $z_{(0)} = 0$ and $z_{(n+1)} = 1$. The maximum gap is

$$G(B) = \max_{0 \leq i \leq n} (z_{(i+1)} - z_{(i)}).$$

Table 8: **BIC comparison between 1PL and 2PL across all benchmarks.** Lower BIC is preferred. We define $\Delta\text{BIC} = \text{BIC}_{2\text{PL}} - \text{BIC}_{1\text{PL}}$, so positive values favor 1PL. BIC selects 1PL for 21 out of 25 benchmarks.

Benchmark	$\text{BIC}_{1\text{PL}}$	$\text{BIC}_{2\text{PL}}$	ΔBIC	Selected
ScienceQA	126817	151726	+24909	1PL
MMLU-Pro-Physics	44290	52569	+8279	1PL
MedQA	52414	60340	+7926	1PL
ZebraLogic	22000	29319	+7319	1PL
MMLU-Pro-Chemistry	37446	44645	+7199	1PL
MMLU-Pro-Biology	21712	25399	+3687	1PL
IFEval	17614	20286	+2672	1PL
USACO	5349	7500	+2151	1PL
BBH-3 objects	5362	6958	+1596	1PL
MBPP+	14222	15778	+1556	1PL
BBH-7 objects	6838	8260	+1422	1PL
BBH-5 objects	6515	7910	+1395	1PL
GAIA	7130	8166	+1036	1PL
Minerva-Math	11865	12798	+933	1PL
GPQA-Diamond	8080	8739	+659	1PL
MATH-500	17194	17739	+545	1PL
OlympiadBench	25166	25466	+300	1PL
SWE-bench-Mini	1624	1876	+252	1PL
CORE-Bench Hard	2881	3106	+225	1PL
HumanEval	5188	5376	+188	1PL
AIME26	1358	1362	+4	1PL
Gaokao2023en	15054	14977	-77	2PL
AIME25	1593	1512	-81	2PL
AIME24	1496	1409	-87	2PL
AMC23	2607	2052	-555	2PL

Table 9: **Posterior diagnostics for benchmarks where BIC favors 2PL.** We report the maximum $\hat{R}$, minimum bulk ESS, and number of divergent transitions across fitted parameters. Although BIC favors 2PL on these four benchmarks, the 2PL fits exhibit substantially worse sampling diagnostics, while the corresponding 1PL fits remain well-behaved.

Benchmark	1PL			2PL		
	Max $\hat{R} \downarrow$	Min Bulk ESS $\uparrow$	# Div. $\downarrow$	Max $\hat{R} \downarrow$	Min Bulk ESS $\uparrow$	# Div. $\downarrow$
Gaokao2023en	1.01	569	0	1.06	63	983
AIME25	1.01	246	0	1.02	188	18
AIME24	1.02	272	0	1.04	118	2
AMC23	1.01	334	0	1.03	146	8

C.5 Model Comparison for Different Difficulty Estimation Models

Our empirical analysis relies on a scalar 1PL IRT difficulty estimate. To assess whether this modeling choice is supported by the response data, we compare 1PL against more flexible IRT specifications, including 2PL and multidimensional IRT models.

We first compare 1PL and 2PL using Bayesian Information Criterion (BIC), where lower BIC is preferred. As shown in Tab. 8, BIC selects 1PL for 21 out of the 25 benchmarks, often by large margins. This indicates that for most benchmarks, adding item-specific discrimination parameters is not supported after accounting for model complexity.

For the remaining four benchmarks where BIC favors 2PL, the 2PL fits exhibit problematic sampling diagnostics, while the corresponding 1PL fits remain well-behaved. Tab. 9 reports the convergence diagnostics for these cases. For example, on Gaokao2023en, the 2PL fit produces max $\hat{R} = 1.06$, min bulk ESS = 63, and 983 divergent transitions, whereas the 1PL fit has max $\hat{R} = 1.01$, min bulk ESS = 569, and zero divergent transitions. Similar issues appear for AIME24, AIME25, and AMC23. Therefore, although 2PL is occasionally favored by BIC, these 2PL fits do not provide reliable alternative difficulty estimates.

Table 10: **Convergence diagnostics for multidimensional IRT fits on smaller benchmarks.** We report the maximum $\hat{R}$, minimum bulk ESS, and number of divergent transitions across fitted parameters. The multidimensional fits generally exhibit poor mixing, with high maximum $\hat{R}$ and very small minimum bulk ESS, indicating that they do not provide stable alternative difficulty estimates.

Benchmark	MIRT-2			MIRT-3		
	Max $\hat{R}$ ↓	Min Bulk ESS ↑	# Div. ↓	Max $\hat{R}$ ↓	Min Bulk ESS ↑	# Div. ↓
AIME24	1.02	202	0	1.68	6	0
AIME25	1.75	6	0	1.02	172	0
AIME26	1.54	7	0	1.44	8	0
AMC23	1.75	6	0	1.35	10	0
Minerva-Math	1.73	6	0	1.37	10	0
GPQA-Diamond	1.81	6	0	1.67	6	0
HumanEval	1.58	7	0	1.90	6	0
MBPP+	2.24	5	0	1.88	6	0
BBH-3 objects	1.78	6	0	1.26	12	0
BBH-5 objects	1.92	6	0	1.23	13	0
BBH-7 objects	1.84	6	0	1.16	20	0
CORE-Bench Hard	1.77	6	0	1.09	31	0
GAIA	1.62	7	0	1.27	12	0
SWE-bench-Mini	1.48	8	0	1.13	20	0
USACO	1.52	7	0	1.35	10	0

We also evaluate multidimensional IRT models with two and three latent dimensions, denoted MIRT-2 and MIRT-3. Tab. 10 reports convergence diagnostics on the 15 smaller benchmarks. Overall, these multidimensional fits do not provide stable posterior estimates: most runs have maximum $\hat{R}$ well above the standard convergence threshold of 1.05 and extremely small minimum bulk ESS, indicating poor mixing. On larger benchmarks, the multidimensional models are substantially more expensive and did not complete reliably within our fitting budget under the same Bayesian fitting procedure. Thus, the current response matrices do not support stable multidimensional difficulty estimates for our empirical analysis.

Overall, these model-comparison results support the use of 1PL as a reliable difficulty estimation model for our study. While more complex IRT specifications may be useful in other settings, our diagnostics indicate that 1PL provides the most stable and well-supported difficulty estimates for the response matrices analyzed here.

C.6 Posterior Uncertainty Analysis

We further quantify how posterior uncertainty in the fitted IRT difficulty estimates affects our calculated metrics. For each posterior draw of the fitted 1PL item difficulties, we recompute DDM and Max Gap, and summarize the posterior variability of these metrics across draws.

DDM is stable under IRT posterior uncertainty. Across the 25 benchmarks, the posterior standard deviation of DDM is typically about 0.011, with range 0.006–0.020. This is small relative to the DDM point estimates, corresponding to approximately 8%–12% in typical cases. Thus, the DDM values reported in the main paper are not artifacts of difficulty-estimation noise.

Max Gap uncertainty is larger but the coverage-gap conclusion remains robust. Max Gap has larger posterior uncertainty, as expected, because it is an extreme-order statistic [11]: it is defined by the largest adjacent spacing between item difficulties and is therefore more sensitive to local posterior perturbations. Its posterior standard deviation ranges from 0.017 to 0.041. However, the qualitative coverage-gap conclusion remains robust. To verify this, we compare the observed Max Gap against a same-size uniform-sampling baseline under posterior difficulty draws. Across benchmarks, the posterior probability that the observed Max Gap exceeds the same-size uniform baseline has median 0.999. The observed-to-baseline ratio has median $3.85\times$, and is much larger for several large benchmarks, e.g., $18.8\times$ on ScienceQA, $11.3\times$ on MedQA, and $5.9\times$ on ZebraLogic. Therefore, although the exact Max Gap magnitude carries posterior uncertainty, the conclusion that many benchmarks contain coverage gaps far beyond random uniform sampling is robust.

C.7 Additional Experimental Details and Results

Our code repository is available at:

<https://anonymous.4open.science/r/Benchmark-Empirical-Measure-527B>.

Algorithm 1 Quantile-Matching Benchmark Calibration

Require: Normalized item difficulties $z_1, \dots, z_n \in [0, 1]$, subset size k , target quantile function Q_μ

- 1: Initialize selected set $A \leftarrow \emptyset$ and available set $U \leftarrow [n]$
- 2: **for** $t = 1, \dots, k$ **do**
- 3: Set target quantile level

$$\tau_t \leftarrow \frac{t - \frac{1}{2}}{k}$$

- 4: Set target difficulty location

$$q_t \leftarrow Q_\mu(\tau_t)$$

- 5: Select the closest unused item:

$$i_t \in \arg \min_{i \in U} |z_i - q_t|$$

- 6: Update $A \leftarrow A \cup \{i_t\}$
 - 7: Update $U \leftarrow U \setminus \{i_t\}$
 - 8: **end for**
 - 9: **return** A
-

We conducted all experiments on NVIDIA L40S GPUs with 48GB memory and NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs with 96GB memory. Fig. 5 provides the difficulty-distribution plots for the remaining benchmarks not shown in Fig. 3, following the same visualization format.

D Benchmark Calibration

Problem setting. The measure-mismatch perspective suggests a natural way to redesign a benchmark: adjust the benchmark empirical measure so that it better matches a target evaluation distribution. Consider a benchmark

$$B = \{x_1, \dots, x_n\},$$

where each item x_i has an estimated difficulty $z_i \in [0, 1]$. The benchmark induces the empirical measure

$$\nu_B = \frac{1}{n} \sum_{i=1}^n \delta_{z_i}$$

over the difficulty axis. Given a target distribution $\mu \in \mathcal{P}([0, 1])$, benchmark calibration aims to construct a new benchmark B' whose empirical measure $\nu_{B'}$ is closer to μ than ν_B is.

In this work, we instantiate benchmark calibration through pruning. Specifically, we retain k items from the original benchmark by selecting a subset $A \subseteq [n]$ with $|A| = k$. The selected items form a calibrated benchmark

$$B_A = \{x_i : i \in A\},$$

with empirical difficulty measure

$$\nu_A = \frac{1}{k} \sum_{i \in A} \delta_{z_i}.$$

The goal is to choose A so that ν_A better approximates the target distribution μ , for example as measured by $\text{DDM}(B_A; \mu) = W_1(\nu_A, \mu)$.

A simple calibration algorithm. We use a simple greedy quantile-matching pruning procedure, summarized in Alg. 1. Let Q_μ denote the quantile function of the target distribution μ . Given a subset size k , we first construct k target difficulty locations by taking the midpoints of k equal-probability intervals under μ :

$$\tau_t = \frac{t - \frac{1}{2}}{k}, \quad q_t = Q_\mu(\tau_t), \quad t = 1, \dots, k.$$

We then match these target locations to existing benchmark items. Specifically, after normalizing item difficulties to $[0, 1]$, we iterate over $q_1, \dots, q_k$ and select the unused item whose difficulty is closest to the current target location:

$$i_t \in \arg \min_{i \in [n] \setminus \{i_1, \dots, i_{t-1}\}} |z_i - q_t|.$$

The calibrated subset is $A = \{i_1, \dots, i_k\}$.

D.1 Experimental Setup

In practice, given any target distribution over the difficulty axis, our benchmark calibration idea and Alg. 1 can be applied directly. In this experiment, we study whether such calibration can make a single benchmark produce model rankings that are more consistent with a broader in-domain benchmark family.

We consider $N = 8$ math benchmarks, as shown in Tab. 3. For each benchmark B_i , we treat it as the source benchmark to be calibrated, and use the remaining benchmarks $\{B_j : j \neq i\}$ only to construct an external reference ranking over models. This leave-one-out design evaluates whether the ranking induced by a calibrated version of B_i better agrees with the ranking obtained from other benchmarks in the same domain.

Concretely, let

$$B_i = \{x_{i1}, \dots, x_{im_i}\}$$

be the i -th benchmark, where item $x_{i\ell}$ has normalized difficulty $z_{i\ell} \in [0, 1]$. Since the leave-one-out reference ranking is aggregated from multiple benchmarks in the same domain, it is less tied to the idiosyncratic difficulty concentration of any single source benchmark. We therefore use calibration to prune items from difficulty regions where the source benchmark is overly concentrated. In the absence of an application-specific target distribution, we instantiate the calibration target as the uniform distribution

$$\lambda = \text{Unif}[0, 1].$$

This choice should not be interpreted as assuming that the leave-one-out reference distribution is exactly uniform. Rather, uniform calibration provides a neutral debiasing strategy: it does not privilege any difficulty region a priori, and it counteracts benchmarks whose empirical mass is concentrated in a narrow portion of the difficulty axis.

We apply Alg. 1 to B_i with target distribution λ , producing a calibrated subset $A_i \subseteq [n_i]$ of size k_i . The calibrated benchmark is

$$B_{i,A_i} = \{x_{i\ell} : \ell \in A_i\}.$$

Suppose all benchmarks evaluate the same M models. Let $S_i(m)$ denote the average score of model m on the full source benchmark B_i , and let $S_i(A_i, m)$ denote its average score on the calibrated subset. The uncalibrated ranking is induced by

$$S_i(m), \quad m = 1, \dots, M,$$

while the calibrated ranking is induced by

$$S_i(A_i, m), \quad m = 1, \dots, M.$$

For the leave-one-out reference ranking, we aggregate model scores over the remaining benchmarks:

$$S_{-i}^{\text{ref}}(m) = \frac{1}{N-1} \sum_{j \neq i} S_j(m).$$

The ranking induced by $S_{-i}^{\text{ref}}(m)$ serves as the external reference ranking for source benchmark B_i .

For each source benchmark, we compute the Spearman and Kendall correlations between the source-induced ranking and the leave-one-out reference ranking, both before and after calibration. Improved correlation after calibration suggests that pruning the source benchmark toward a more balanced difficulty distribution makes its model comparison more consistent with the broader in-domain benchmark family.

D.2 Results

Experimental results are reported in Tab. 11. The results show that our calibration algorithm improves the agreement between the source benchmark ranking and the leave-one-out reference ranking. On average, calibration improves Spearman correlation from 0.890 to 0.930, and Kendall correlation from 0.802 to 0.848. The improvement is especially pronounced on benchmarks with weaker initial agreement. Notably, on Minerva-Math, the full benchmark achieves only 0.601 Spearman and 0.455 Kendall correlation with the reference ranking, while calibration improves them to 0.790 and 0.667, respectively. Similarly, on Gaokao2023en, calibration improves Spearman from 0.888 to 0.951 and Kendall from 0.788 to 0.879. For the remaining benchmarks, the initial agreement is already high, leaving less room for improvement; nevertheless, calibration consistently preserves this high level of agreement while using the pruned benchmark.

We include a random pruning baseline that uses the same pruning size as our calibrated subset and average its results over 30 random trials. Random pruning is less effective than calibration: it achieves 0.895 average Spearman and 0.801 average Kendall correlation, both below the calibrated subset. We also report $W_1(\nu, \text{Unif})$ as a mechanism metric. Calibration consistently reduces this distance across all benchmarks, decreasing the average value from 0.096 to 0.028. In contrast, random pruning does not reduce this distance on average, yielding an average value of 0.109. This suggests that the improvement is not simply due to using a smaller benchmark, but comes from selecting items whose empirical difficulty measure better matches the neutral calibration target.

Table 11: **Benchmark calibration results on eight math benchmarks.** For each source benchmark, **Full** denotes the original benchmark, **Random** denotes the average over 30 random pruned subsets of the same size, and **Cal.** denotes the calibrated subset produced by our quantile-matching benchmark calibration algorithm. Spearman and Kendall measure agreement with the leave-one-out reference ranking. We report $W_1(\nu, \text{Unif})$ as a mechanism metric to verify that calibration moves the selected benchmark measure closer to the neutral uniform distribution. Bold values indicate the best result among **Full**, **Random**, and **Cal.** for each benchmark and metric.

Benchmark	Spearman $\uparrow$			Kendall $\uparrow$			$W_1(\nu, \text{Unif}) \downarrow$		
	Full	Random	Cal.	Full	Random	Cal.	Full	Random	Cal.
Minerva-Math	0.601	0.649	0.790	0.455	0.514	0.667	0.122	0.115	0.005
MATH-500	0.930	0.942	0.958	0.879	0.878	0.909	0.164	0.165	0.099
Gaokao2023en	0.888	0.895	0.951	0.788	0.790	0.879	0.088	0.086	0.029
AIME24	0.944	0.942	0.944	0.848	0.847	0.848	0.077	0.108	0.019
AIME25	0.958	0.936	0.972	0.909	0.854	0.915	0.079	0.126	0.020
AIME26	0.949	0.950	0.970	0.870	0.867	0.901	0.085	0.108	0.017
AMC23	0.937	0.934	0.937	0.848	0.848	0.848	0.088	0.093	0.025
OlympiadBench	0.916	0.910	0.916	0.818	0.806	0.818	0.064	0.067	0.008
Average	0.890	0.895	0.930	0.802	0.801	0.848	0.096	0.109	0.028

E Additional Details and Experimental Results of BENCHFILL

Algorithm 2 BENCHFILL

Require: Benchmark items with normalized difficulties $\{(x_i, z_i)\}_{i=1}^n$; target gap $[z_L, z_R]$; rewriter LLM $\mathcal{G}$; validity judge $\mathcal{J}$; IRT evaluation LLMs $\{M_j\}_{j=1}^m$ with pre-estimated IRT abilities $\{\hat{\theta}_j\}_{j=1}^m$; trials per LLM $\{K_j\}_{j=1}^m$; number of parents R ; children per parent C ; reference list budget L ; maximum rounds T ; tolerance τ on the normalized $[0, 1]$ scale.

Ensure: A proposed gap-filling item x_{item}^* .

- 1: Set target difficulty $z^* \leftarrow (z_L + z_R)/2$.
- 2: Initialize memory $\mathcal{M}$ with existing annotated item–difficulty pairs near the selected gap.
- 3: Initialize valid candidate set $\mathcal{C}_{\text{valid}} \leftarrow \emptyset$.
- 4: **for** $t = 1, \dots, T$ **do**
- 5: Retrieve a parent set $\mathcal{S}_t \subseteq \mathcal{M}$ with at most R items whose annotated difficulties are closest to z^* .
- 6: Construct a reference list $\mathcal{R}_t \subseteq \mathcal{M}$ with $|\mathcal{R}_t| \leq L$ using nearby item–difficulty pairs around z^* , with up to half selected from below z^* and the rest from above z^* when available.
- 7: **for** each parent item $(x_p, z_p) \in \mathcal{S}_t$ **do**
- 8: Rescale the displayed difficulties in z^* and $\mathcal{R}_t$ from $[0, 1]$ to $[0, 10]$ for prompting.
- 9: Rewrite the parent into C child items conditioned on the parent item, target difficulty, and reference list:

$$\{x_{p,1}, \dots, x_{p,C}\} \leftarrow \mathcal{G}(x_p, z^*, \mathcal{R}_t).$$

- 10: **for** each child item $x_{p,c}$ **do**
- 11: **if** $\mathcal{J}$ verifies that the reference answer of $x_{p,c}$ is correct **then**
- 12: **for** $j = 1, \dots, m$ **do**
- 13: Evaluate LLM M_j on $x_{p,c}$ for K_j trials and record c_j correct responses.
- 14: **end for**
- 15: Estimate the item difficulty by online calibration:

$$\hat{b}_{p,c} \leftarrow \arg \max_b \sum_{j=1}^m \left[c_j \log \sigma(\hat{\theta}_j - b) + (K_j - c_j) \log(1 - \sigma(\hat{\theta}_j - b)) \right].$$

- 16: Normalize $\hat{b}_{p,c}$ using the same transformation as the original benchmark and obtain $\hat{z}_{p,c}$.
- 17: Add $(x_{p,c}, \hat{z}_{p,c})$ to $\mathcal{M}$.
- 18: Add $(x_{p,c}, \hat{z}_{p,c})$ to $\mathcal{C}_{\text{valid}}$.
- 19: **end if**
- 20: **end for**
- 21: **end for**
- 22: **if** $\mathcal{C}_{\text{valid}} \neq \emptyset$ and $\min_{(x,\hat{z}) \in \mathcal{C}_{\text{valid}}} |\hat{z} - z^*| \leq \tau$ **then**
- 23: **break**
- 24: **end if**
- 25: **end for**
- 26: **if** $\mathcal{C}_{\text{valid}} = \emptyset$ **then**
- 27: **return** failure
- 28: **end if**
- 29: Select the closest valid child item:

$$(x_{\text{item}}^*, \hat{z}^*) \leftarrow \arg \min_{(x,\hat{z}) \in \mathcal{C}_{\text{valid}}} |\hat{z} - z^*|.$$

- 30: **return** x_{item}^* as the proposed gap-filling item, subject to human expert review.
-

E.1 Implementation Details

For each benchmark, we focus on the three most pronounced gaps in its estimated difficulty distribution and rewrite one item for each gap. Since an LLM may not reliably distinguish small numerical differences on the normalized $[0, 1]$ difficulty scale when the target difficulty is provided in the prompt, we rescale difficulty values to $[0, 10]$ for prompting while keeping all reported metrics on the original normalized scale.

For the reference list, we include nearby item–difficulty pairs on both sides of the target difficulty. Specifically, half of the reference items are selected from those with difficulty below the target and closest to it, and the other

half are selected from those with difficulty above the target and closest to it. This provides the rewriter with local bidirectional context around the desired difficulty level. The full rewriter prompt template is provided below.

For difficulty annotation, we use the same set of LLMs and the same generation configuration as in the original difficulty estimation stage. The search stops early if the best valid item found so far is within tolerance $\tau = 0.01$ of the target difficulty on the normalized $[0, 1]$ scale, or otherwise stops after the maximum number of rounds. We then select the valid item whose annotated difficulty is closest to the target. Before being inserted into the benchmark, each selected item undergoes a manual validity check by at least three human experts, who verify its correctness.

We additionally implement a budget-matched best-of- N uniform-parent rewriting baseline. For each target gap g , let T_g denote the number of rounds actually run by BENCHFILL before stopping. Since BENCHFILL rewrites 4 parents per round and produces 4 child items per parent, its proposal budget for this gap is $T_g \times 4 \times 4 = 16T_g$ child items. To match this budget, the baseline uniformly samples 4 existing benchmark items as parents and asks the same rewriter LLM to rewrite $4T_g$ child items from each parent, conditioned on the parent item and the target difficulty. After difficulty annotation, it selects the valid child item whose difficulty is closest to the target among all generated candidates. Thus, the baseline uses the same total rewrite budget and the same final selection rule as BENCHFILL, but does not use difficulty-aware parent retrieval, nearby reference lists, or iterative memory updates.

Rewriter Prompt Template

```
Rewrite the parent item below into exactly 1 new benchmark item variant. Target
difficulty (0-10 scale, higher = harder): {target_difficulty}.

Parent item (rewrite this item into a new variant - do not copy its specific
wording, numbers, or answer):
{parent.question}
Answer: {parent.reference_answer}
Difficulty (0-10): {parent.difficulty} | delta {target - parent.difficulty} |
generate {harder|easier}

Memory items (use as style/difficulty reference only, do not copy):

-- Memory 1 --
{memory[0].question}
Answer: {memory[0].reference_answer}
Difficulty (0-10): {memory[0].difficulty} | delta {target -
memory[0].difficulty} | generate {harder|easier}

-- Memory 2 --
{memory[1].question}
Answer: {memory[1].reference_answer}
Difficulty (0-10): {memory[1].difficulty} | delta {target -
memory[1].difficulty} | generate {harder|easier}

... (one block per memory item) ...

Return only one markdown code block: start the first line with ```json, then a
newline, then a single JSON array with exactly 1 element, then close with ``` on
its own line. Do not put anything after the closing fence.

Do not stop until the JSON is syntactically complete: every string value is
closed with ", every object is closed with }, the array ends with ], and then
the closing ``` line.

The array must have exactly 1 element - an object with exactly these keys: "id"
(string), "question" (string), "answer" (string).
```

E.2 Additional Experimental Results

Human validation. We ensure the reliability of the generated gap-filling items using a two-stage validation process. GPT-5.4 first serves as an automatic validity judge, checking whether each candidate is well-formed and whether its reference answer is correct; candidates that fail this check are discarded and regenerated. Human experts then independently audit the final selected items. Across the 24 target gaps, one initially proposed candidate failed the GPT-5.4 check because of an incorrect reference answer and was regenerated. All 24 final selected items passed the GPT-5.4 validity check and subsequently passed human review without modification or replacement.

Table 12: **Localized-improvement detection results.** Each augmentation method adds one item to the original 30-item benchmark. Gain denotes the change in average accuracy. The one-sided stratified randomization test randomly reassigns rollout outcomes between the pre- and post-SFT conditions within each item using 100,000 permutations. Boldface indicates variants for which a positive gain is detected at $p < 0.05$.

Case	Benchmark variant	Gain	One-sided p	Detected
AIME24 Olmo-3.1-32B-Think	Original	+0.104	0.500100	×
	Original + BenchFill	+2.419	0.005940	✓
	Original + Baseline	+0.202	0.458000	×
AIME25 Phi-4	Original	+1.042	0.184100	×
	Original + BenchFill	+3.730	0.000450	✓
	Original + Baseline	+0.907	0.214200	×
AIME26 Llama-3.1-8B-Instruct	Original	+0.208	0.418400	×
	Original + BenchFill	+3.427	0.000010	✓
	Original + Baseline	+0.101	0.498000	×

Does the rewriter perceive difficulty? A natural question is whether our rewriter indeed perceives difficulty, instead of hitting the target difficulty blindly. We examine this question from two complementary angles: whether the rewriter already shows difficulty sensitivity before iterative feedback, and whether the subsequent search-and-annotate loop further improves alignment with the target difficulty.

First, to isolate the rewriter’s own difficulty perception from later IRT feedback and iterative search, we analyze only the first generation round of each gap-filling case. At this stage, the rewriter is conditioned on the specified target difficulty, but has not yet received feedback on the annotated difficulties of its generated children. Fig. 6 shows the specified target difficulty and the annotated difficulty of valid first-round children across all 8 benchmarks and 3 target gaps per benchmark. Within most benchmarks, higher specified targets produce higher mean annotated difficulties, and well-separated targets usually induce distinct child-difficulty distributions. This indicates that the rewriter already has non-trivial sensitivity to the specified difficulty scale before iterative feedback is introduced.

Second, we examine whether the search-and-annotate loop further improves target alignment. Fig. 7 shows one representative gap-filling case from each benchmark. Across these cases, the mean absolute target deviation generally decreases from the first round to the final round, and in most cases the generated children also become more concentrated around the target. This suggests that the iterative loop further refines the rewriter’s initial difficulty signal using IRT annotation and updated memory.

Detecting localized model improvements. We further examine whether BENCHFILL improves a benchmark’s ability to detect localized model improvements beyond reducing difficulty gaps. We consider one target gap in each of AIME24, AIME25, and AIME26, and compare three benchmark variants: `Original`, `Original + BenchFill`, and `Original + Baseline`. Both augmentation methods insert exactly one item into the original benchmark. We perform mild SFT so that the model improves around the target gap region. Each item is evaluated with 32 rollouts before and after SFT, and we test whether the gain is statistically detectable using a one-sided stratified randomization test ($p < 0.05$).

As shown in Tab. 12, `Original + BenchFill` detects a positive post-SFT gain in all three cases (3/3), while `Original` and `Original + Baseline` detect none (0/3 each). This shows that BENCHFILL is not merely improving the metric it optimizes; by filling under-covered difficulty regions, it makes the benchmark more sensitive to localized model improvements that would otherwise be missed.

Representative examples. To provide a qualitative view of the generated gap-filling items, we show representative rewriting examples in Tab. 13. Each example presents a final selected item together with its original ancestor from the benchmark, as well as the ancestor difficulty, target difficulty, and annotated difficulty of the selected item. The examples illustrate how BENCHFILL modifies existing benchmark items to reach previously under-covered difficulty levels.

Table 13: **Representative pairs of original benchmark ancestors and final selected items produced by BENCHFILL.** All difficulty values are normalized to $[0, 1]$. The main change from the ancestor item to the selected item is highlighted. For selected items, values in parentheses denote the target difficulty.

Benchmark	Original benchmark ancestor		Selected gap-filling item	
	Item	Diff.	Item	Diff.
AMC23	For how many integers n does the expression $\sqrt{\frac{\log(n^2) - (\log n)^2}{\log n - 3}}$ represent a real number, where $\log$ denotes the base-10 logarithm?	0.535	Find the number of integer values of n such that $\sqrt{\frac{\log(n^2) - (\log n)^2}{\log n - 1}}$ represents a real number, where $\log$ denotes the base-10 logarithm.	0.597 (0.598)
Gaokao2023en	On an 8×8 chessboard, 6 black rooks and k white rooks are placed on different cells so that each rook only attacks rooks of the opposite color. Compute the maximum possible value of k . (Two rooks attack each other if they are in the same row or column and no rooks are between them.)	0.864	On a 9×9 chessboard, 7 black rooks and k white rooks are placed on distinct squares so that each rook attacks only rooks of the opposite color. What is the maximum possible value of k ? (Two rooks attack each other if they are in the same row or column and there is no rook between them.)	0.886 (0.885)
AIME25	Alex divides a disk into four quadrants with two perpendicular diameters intersecting at the center of the disk. He draws 25 more line segments through the disk, each connecting two points at random on the perimeter in different quadrants. Find the expected number of regions into which these 27 line segments divide the disk.	0.750	A circle has a diameter drawn, splitting it into two semicircular regions. Then 8 chords are drawn, each connecting a random point on the boundary in one semicircle to a random point on the boundary in the other semicircle. Assuming all points are chosen uniformly and independently, what is the expected number of regions into which the interior of the circle is divided?	0.674 (0.671)

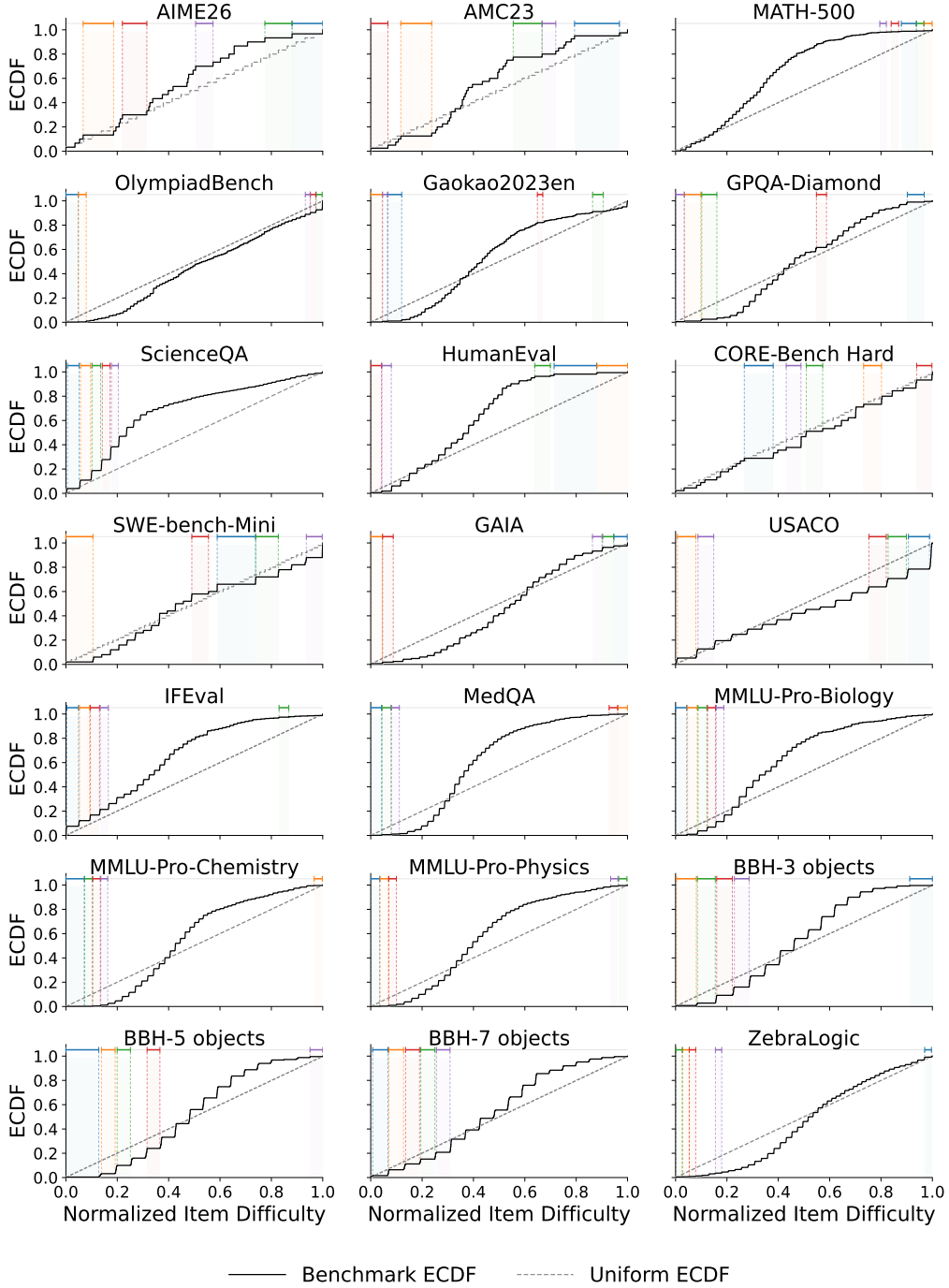


Figure 5: **Additional benchmark difficulty distributions.** Each panel shows the ECDF of normalized item difficulties for one benchmark. The black curve denotes the benchmark ECDF, and the gray dashed line denotes the uniform ECDF, which we use as the target difficulty distribution. Deviations from the uniform ECDF indicate mismatch relative to this neutral target distribution. Colored shaded intervals mark the five largest adjacent gaps between sorted item difficulties.

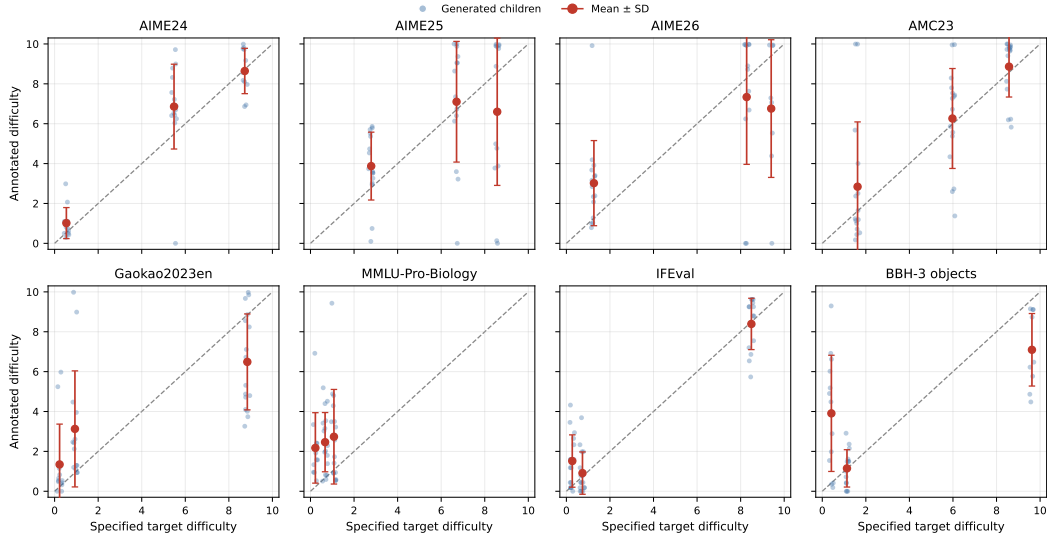


Figure 6: **First-round alignment of specified target difficulty and annotated difficulty.** Each panel corresponds to one benchmark. The horizontal axis shows the specified target difficulty, and the vertical axis shows the annotated difficulty of valid first-round children. Blue points denote individual generated children, and red markers denote the mean and standard deviation for each target difficulty. The dashed diagonal indicates perfect alignment. For readability, both axes are shown on a 0–10 scale, corresponding to normalized difficulties in $[0, 1]$ multiplied by 10. Within most benchmarks, higher specified targets produce higher annotated difficulties already in the first round, before any iterative feedback is incorporated.

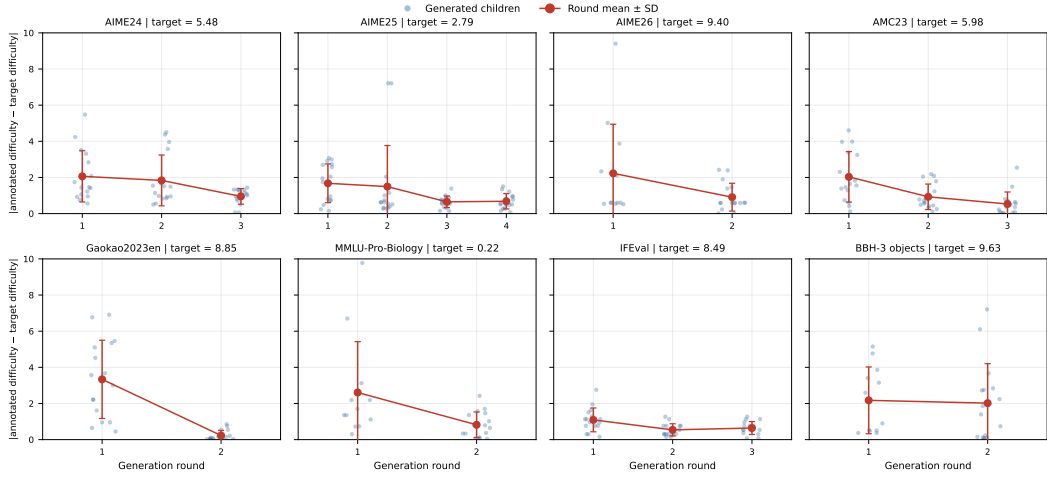


Figure 7: **Multi-round target alignment during BENCHFILL search.** Each panel shows one representative gap-filling case from a different benchmark. The horizontal axis denotes the generation round, and the vertical axis denotes the absolute target deviation, $|\hat{z} - z^*|$, for generated children. Blue points denote individual generated children, and red markers denote the round-level mean and standard deviation. Across most cases, the mean target deviation decreases over rounds, and the generated children become more concentrated around the target, showing that the search-and-annotate loop further improves target alignment beyond the rewriter’s initial difficulty sensitivity.