AN INTERPRETABLE AND SCALABLE FRAMEWORK FOR EVALUATING LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluation of large language models (LLMs) is increasingly critical, yet standard benchmarking methods rely on average accuracy, overlooking both the inherent stochasticity of LLM outputs and the heterogeneity of benchmark items. Item Response Theory (IRT) offers a principled framework for modeling latent model abilities and item characteristics, but conventional methods are computationally expensive and numerically unstable, limiting large-scale implementations. To address these challenges, we propose an interpretable and scalable framework for LLM evaluation based on the majorization–minimization principle. Our approach reformulates the problem as a sequence of constrained matrix factorization subproblems, enabling stable and efficient parameter estimation with theoretical guarantees for identifiability and convergence. Experiments on synthetic and real-world datasets, including MATH-500 and six Open LLM Leaderboard benchmarks, demonstrate that our method achieves superior scalability and interpretability. It delivers orders-of-magnitude speedups over competing methods while maintaining comparable or even higher estimation accuracy. Our results align with established scaling laws and provide interpretable insights into item difficulty and discrimination, informing more principled benchmark design than conventional average-accuracy evaluation.

1 INTRODUCTION

Given the rapid evolution of large language model (LLM) systems and their expanding applications (Chen et al., 2024; Dong et al., 2025; Lin et al., 2025), rigorous evaluation of LLMs is indispensable. Benchmarks have been the gold standard for developers to demonstrate model capabilities (Achiam et al., 2024; Comanici et al., 2025), with performance typically measured by average accuracy between ground-truth labels and model-generated outputs across all benchmark items. However, this conventional approach overlooks two fundamental challenges: the stochastic nature of LLM outputs and the heterogeneity of benchmark items. First, LLM outputs can vary substantially under minor perturbations, such as changes in prompt order, slight typos, or the use of synonyms (Nie et al., 2020; Lu et al., 2022; Zhu et al., 2024). Under greedy decoding settings, outputs can be inconsistent even when presented with identical prompts (Atıl et al., 2025), leading to fragile evaluations and complicating subsequent interpretation. Second, benchmark items exhibit substantial heterogeneity in their evaluative utility, in contrast to the implicit equal-weight assumption underlying average accuracy. Highly discriminative items can better differentiate model capabilities and thus warrant greater weights (Burnell et al., 2023), whereas items compromised by training data contamination lose their evaluative integrity and should be downweighted (Li, 2023). Furthermore, as benchmarks evolve, some items become obsolete or redundant (Yu et al., 2024), which reduces evaluation efficiency and warrants exclusion.

These challenges call for evaluation frameworks that can recover intrinsic model abilities under stochastic generation while capturing heterogeneity across benchmark items. Recently, psychometricians have brought Item Response Theory (IRT) into the picture (Castleman et al., 2025; Zhuang et al., 2025; Xu et al., 2025), framing LLM evaluation as a latent variable modeling problem analogous to human performance assessment. IRT models each binary evaluation outcome as a sample drawn from a latent distribution, parameterized by both model traits and item features. Estimating these parameters enables a systematic decomposition of model performance, moving beyond average accuracy toward a more interpretable and diagnostically informative evaluation paradigm.

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

However, traditional IRT-based evaluation methods (Chalmers, 2012; Xu et al., 2025) that rely on the expectation-maximization (EM) algorithm (Dempster et al., 1977) suffer from critical limitations, most notably **computational inefficiency** and **numerical instability**. These methods are often prohibitively slow (Wu et al., 2020; Lalor & Rodriguez, 2023), and their performance is extremely sensitive to singular data configurations, such as near-constant responses and collinearity (Baker & Kim, 2004; Chen et al., 2025). Such issues are particularly pronounced in LLM evaluation settings, where training data leakage or extremely difficult items frequently arise (Li, 2023; Chai et al., 2026). As a result, existing IRT approaches exhibit **poor scalability** and remain largely restricted to relatively **small-scale applications**, making them ill-suited for the rapidly growing ecosystem of LLMs and benchmark datasets.

To address these limitations, we propose an interpretable and scalable computational framework for IRT-based LLM evaluation by employing the majorization-minimization (MM) principle (Lange et al., 2000; Hunter & Lange, 2004). Specifically, we reformulate the original problem as a sequence of constrained matrix factorization subproblems, thereby stabilizing and accelerating the estimation procedure. We further address parameter identifiability and establish theoretical guarantees for algorithmic convergence. Our primary contributions are as follows:

- **A novel MM-based framework**: We develop an interpretable and scalable computational framework for LLM evaluation grounded in MM optimization, well-suited to large-scale evaluation datasets and capable of offering reliable and meaningful interpretations.
- **Theoretical guarantees**: We provide theoretical guarantees for parameter identifiability and algorithmic convergence, ensuring reliable recovery of model abilities and item features.
- **Computational efficiency & scalability**: Extensive large-scale experiments demonstrate orders-of-magnitude speedup over competing methods while achieving comparable or even superior estimation accuracy.
- **Real-world applications & empirical insights**: We evaluate on MATH-500 and six benchmarks from the Hugging Face Open LLM Leaderboard. Results on model abilities are consistent with established parametric scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022), and the recovered item features align with human annotations (Lightman et al., 2024), capturing fine-grained heterogeneity that average accuracy fails to reflect.

Overall, this work presents a novel MM-based computational framework with rigorous theoretical guarantees, enhanced computational efficiency and scalability, and strong empirical interpretability, showcasing a substantial advancement over conventional IRT-based approaches for LLM evaluation.

2 PRELIMINARY

We first conceptualize the evaluation process within a latent probabilistic framework. Given a set of N LLMs and J benchmark items, let $y_{ij} \in \{-1, 1\}$ denote whether the i-th LLM correctly addressed the j-th benchmark item, yielding a binary response matrix $\mathbf{Y} \in \{-1, 1\}^{N \times J}$. We assume that an LLM's output is governed by an intrinsic stochastic process, modeling the probability of a correct response via a logit link function:

$$\Phi(x_{ij}) \equiv \mathbb{P}\left(y_{ij} = 1 \mid x_{ij}\right) = \frac{1}{1 + \exp\left(-x_{ij}/\sigma\right)}, \tag{1}$$

where x_{ij} represents the pre-activation score produced by the i-th model for the response of item j, and σ acts as the scaling parameter. This formulation mirrors standard next-token decoding, where the multi-class softmax over the vocabulary collapses into a logistic distribution when projected onto the binary correctness space, with σ corresponding directly to the temperature (Guo et al., 2017).

To obtain interpretable representations for fine-grained evaluation, we adopt a reparametrization based on the two-parameter IRT model (2PL-IRT) (Hambleton & Swaminathan, 1985; Chen et al., 2025), expressing the score matrix as

$$\mathbf{X} = \boldsymbol{\theta}\mathbf{a}^\top + \mathbf{1}\mathbf{b}^\top = \begin{pmatrix} \boldsymbol{\theta} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{a}^\top \\ \mathbf{b}^\top \end{pmatrix} \equiv \mathbf{U}\mathbf{V}^\top, \tag{2}$$

where $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_N)^\top$, $\mathbf{a} = (a_1, \cdots, a_J)^\top$, $\mathbf{b} = (b_1, \cdots, b_J)^\top$, and $\mathbf{1} \in \mathbb{R}^N$ is a vector of ones. Each entry x_{ij} is decomposed as $x_{ij} = a_j\theta_i + b_j$, where a_j denotes item discrimination, θ_i represents model ability, and b_j is an item easiness parameter. This formulation reveals a critical coupling: items with larger a_j amplify differences in model ability, yielding stronger discriminative power, whereas items with small a_j provide limited signal for distinguishing model capabilities. Following standard 2PL-IRT frameworks, we impose a_j to be non-negative to ensure a monotonic relationship between model ability and response accuracy, preserving interpretability of the discrimination parameter. The intercept b_j captures item easiness, serving as a baseline for the score that is independent of specific model-item interactions.

3 METHODOLOGY

To recover the triplet representation $(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b})$, a straightforward approach is to minimize a binary cross-entropy loss accounting for all evaluation entries

$$\ell(\mathbf{X}) = -\sum_{(i,j)\in\Omega} \{\gamma_{ij} \log \Phi(x_{ij}) + (1-\gamma_{ij}) \log(1-\Phi(x_{ij}))\}, \tag{3}$$

with $\mathbf{X} = \boldsymbol{\theta}\mathbf{a}^\top + \mathbf{1}\mathbf{b}^\top$, subject to $\mathbf{a} \geq \mathbf{0}$. Here, the scaled response variable is defined as $\gamma_{ij} = \frac{1}{2}(1 + y_{ij})$, and $\Omega \subseteq \{(i,j) \mid 1 \leq i \leq N, 1 \leq j \leq J\}$ denotes the set of observed entries. For $(i,j) \notin \Omega$, we set $y_{ij} = 0$.

3.1 THE MAJORIZATION-MINIMIZATION PRINCIPLE

We employ the MM principle (Lange et al., 2000; Hunter & Lange, 2004) to convert the challenging minimization of $\ell(\mathbf{X})$ into a sequence of simpler surrogate problems. This is achieved through a majorization approximation $g(\mathbf{X} \mid \tilde{\mathbf{X}})$ anchored at the current estimate $\tilde{\mathbf{X}}$, satisfying (i) a tangency condition $g(\tilde{\mathbf{X}} \mid \tilde{\mathbf{X}}) = \ell(\tilde{\mathbf{X}})$ for all $\tilde{\mathbf{X}}$ and (ii) a domination condition $g(\mathbf{X} \mid \tilde{\mathbf{X}}) \geq \ell(\mathbf{X})$ for all $\mathbf{X}$. The two conditions together ensure a monotone decrease of the objective $\ell(\mathbf{X})$ through iterating

$$\mathbf{X}^{(t+1)} = \arg\min_{\mathbf{X}} g(\mathbf{X} \mid \mathbf{X}^{(t)}) \tag{4}$$

for iteration step $t = 0, 1, \cdots$. Notably, exact minimization of the surrogate function is not required to ensure descent of the objective. Therefore, one can inexactly solve (4) and still guarantee a monotonic decrease.

Given the log-likelihood $\log \Phi(x)$ is L-Lipschitz differentiable with $L = 1/4\sigma^2$, and that the density function $\phi(x) = \Phi'(x)$ is symmetric around zero, we invoke Proposition 2.1 in Liu et al. (2025) to derive a quadratic majorization of $\ell(\mathbf{X})$ at $\tilde{\mathbf{X}}$, expressed as

$$g(\mathbf{X} \mid \tilde{\mathbf{X}}) = \frac{L}{2} \left\| \mathbf{X} - \tilde{\mathbf{Y}} \right\|^2_{\mathrm{F}(\Omega)} + c(\tilde{\mathbf{X}}), \tag{5}$$

where $\tilde{\mathbf{Y}} = \tilde{\mathbf{X}} + 4\sigma\mathbf{Y} \circ \Phi(-\mathbf{Y} \circ \tilde{\mathbf{X}})$ under the logit link (1) with $\circ$ denoting the Hadamard product, and $c(\tilde{\mathbf{X}})$ is a constant depending on $\tilde{\mathbf{X}}$. Here $\|\cdot\|_{\mathrm{F}(\Omega)}$ denotes the observed-entry Frobenius seminorm, defined for a matrix $\mathbf{A}$ as $\|\mathbf{A}\|_{\mathrm{F}(\Omega)} = \sqrt{\sum_{(i,j)\in\Omega} a_{ij}^2}$.

3.2 CONSTRAINED BLOCK-WISE OPTIMIZATION

By integrating the reparameterization (2) into the quadratic majorization equation 5, each MM update reduces to solving a constrained matrix factorization problem

$$\min_{\mathbf{U}=(\boldsymbol{\theta},\mathbf{1}),\mathbf{V}=(\mathbf{a},\mathbf{b})} \|\mathbf{U}\mathbf{V}^\top - \tilde{\mathbf{Y}}\|^2_{\mathrm{F}(\Omega)}, \ \text{ subject to } \ \mathbf{a} \geq \mathbf{0}. \tag{6}$$

Observing that the objective function in (6) is quadratic in $\boldsymbol{\theta}$, $\mathbf{a}$, and $\mathbf{b}$, we alternately optimize over these variables. Given $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{b}}$ from the previous iteration, updating $\mathbf{a}$ reduces to a standard entrywise non-negative least squares (NNLS) problem

$$\min_{\mathbf{a}\geq\mathbf{0}} \|\hat{\boldsymbol{\theta}}\mathbf{a}^\top - (\tilde{\mathbf{Y}} - \mathbf{1}\hat{\mathbf{b}}^\top)\|^2_{\mathrm{F}(\Omega)},$$

Algorithm 1 Constrained Block MM (cBMM)

Input: Ω, $\mathbf{Y}$, tolerance tol
Output: $\boldsymbol{\theta}$, $\mathbf{a}$, $\mathbf{b}$
Initialize $\mathbf{U} = (\mathbf{u}, \mathbf{1})$, $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2)$, set $\delta = \infty$
while $\delta >$ tol **do**
  $\tilde{\mathbf{X}} \leftarrow \mathbf{u}\mathbf{v}_1^\top + \mathbf{1}\mathbf{v}_2^\top$
  $\tilde{\mathbf{Y}} \leftarrow \tilde{\mathbf{X}} + 4\sigma\mathbf{Y} \circ \Phi(-\mathbf{Y} \circ \tilde{\mathbf{X}})$
  Compute $\mathbf{R} \leftarrow \tilde{\mathbf{Y}} - \mathbf{1}\mathbf{v}_2^\top$ and $\mathbf{v}_1 \leftarrow \arg\min_{\mathbf{v}_1 \geq \mathbf{0}} \|\mathbf{u}\mathbf{v}_1^\top - \mathbf{R}\|_{\mathrm{F}(\Omega)}^2$
  Update $\mathbf{v}_2 \leftarrow (\tilde{\mathbf{Y}} - \mathbf{u}\mathbf{v}_1^\top)^\top \mathbf{1}/N$
  Update $\mathbf{R} \leftarrow \tilde{\mathbf{Y}} - \mathbf{1}\mathbf{v}_2^\top$ and $\mathbf{u} \leftarrow \mathbf{R}\mathbf{v}_1(\mathbf{v}_1^\top \mathbf{v}_1)^{-1}$
  $\delta \leftarrow |\ell(\mathbf{u}\mathbf{v}_1^\top + \mathbf{1}\mathbf{v}_2^\top) - \ell(\tilde{\mathbf{X}})|/|\ell(\tilde{\mathbf{X}})|$
end while
return $\boldsymbol{\theta} = \mathbf{u}, \quad \mathbf{a} = \mathbf{v}_1, \quad \mathbf{b} = \mathbf{v}_2$

which we solve using the active set method (Lawson & Hanson, 1995). With the updated $\hat{\mathbf{a}}$, the parameter $\mathbf{b}$ admits a closed-form update $\hat{\mathbf{b}} = (\tilde{\mathbf{Y}} - \hat{\boldsymbol{\theta}}\hat{\mathbf{a}}^\top)^\top \mathbf{1}/N$. Subsequently, given $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$, the update for $\boldsymbol{\theta}$ is obtained in closed form as $\hat{\boldsymbol{\theta}} = (\tilde{\mathbf{Y}} - \mathbf{1}\hat{\mathbf{b}}^\top)\hat{\mathbf{a}}/(\hat{\mathbf{a}}^\top\hat{\mathbf{a}})$. Detailed derivations are omitted for brevity. We refer to the proposed computational framework as constrained block MM (cBMM), and summarize the full procedure in Algorithm 1. Per-iteration computational complexity is provided in the Appendices A.4.

3.3 Identifiability

As noted in Hambleton & Swaminathan (1985) and Xu et al. (2025), the estimation of $\boldsymbol{\theta}$ and $\mathbf{a}$ under reparameterization (2) is subject to identifiability issues, such as trivial solutions and symmetries involving sign-flips and scaling, leading to multiple optima with an identical score matrix $\mathbf{X}$. To resolve these ambiguities, specific structural constraints and architectural considerations must be imposed. We begin by providing the formal definition of identifiability.

Definition 1 (Identifiability)**.** *The latent probabilistic model is identifiable if for any solution* $(\boldsymbol{\theta}^*, \mathbf{a}^*, \mathbf{b}^*)$*, another suite of* $(\boldsymbol{\theta}^\dagger, \mathbf{a}^\dagger, \mathbf{b}^\dagger)$ *that induces the same distribution matrix, i.e.,* $\Phi(\mathbf{X}^\dagger) = \Phi(\mathbf{X}^*)$*, must satisfy* $(\boldsymbol{\theta}^\dagger, \mathbf{a}^\dagger, \mathbf{b}^\dagger) = (\boldsymbol{\theta}^*, \mathbf{a}^*, \mathbf{b}^*)$.

Note that strict identifiability based on Definition 1 is not generally guaranteed, as hinted by the ambiguities mentioned previously. We show that, however, the model is identifiable up to an equivalence class, as formalized in Definition 2, with the underlying order structure preserved.

Definition 2 (Equivalence class)**.** *Solution sets* $(\boldsymbol{\theta}^*, \mathbf{a}^*, \mathbf{b}^*)$ *and* $(\boldsymbol{\theta}^\dagger, \mathbf{a}^\dagger, \mathbf{b}^\dagger)$ *are within an equivalence class if each corresponding pair,* $(\boldsymbol{\theta}^*, \boldsymbol{\theta}^\dagger)$*,* $(\mathbf{a}^*, \mathbf{a}^\dagger)$*, and* $(\mathbf{b}^*, \mathbf{b}^\dagger)$*, induces the same rank ordering.*

With all definitions in place, we are ready to present the following theorem to characterize identifiability of (2). We leave the full proof to the Appendices A.2.

Theorem 1 (Structural preservation)**.** *The latent probabilistic model (1) is identifiable up to an equivalence class if* $\boldsymbol{\theta} \neq \mathbf{c}$*, where* $\mathbf{c}$ *is a constant vector, and* $\mathbf{a} \geq \mathbf{0}$ *with* $\mathbf{a} \neq \mathbf{0}$.

3.4 Convergence

We establish convergence guarantees for cBMM under the following assumptions.

(H1) **Boundedness.** The iterates $\{(\boldsymbol{\theta}^{(t)}, \mathbf{a}^{(t)}, \mathbf{b}^{(t)})\}$ are contained in a compact set $\mathcal{K}$, with $\|\mathbf{W}\| \leq M$ for all $\mathbf{W} \in \mathcal{K}$ and some constant $M > 0$.

(H2) **Non-degeneracy.** There exists a constant $c > 0$ such that $\|\boldsymbol{\theta}^{(t)}\| \geq c$ and $\|\mathbf{a}^{(t)}\| \geq c$ for all $t \geq 0$.

Assumption (H1) is standard for iterative optimization algorithms and can be enforced by adding a bounding constraint to the feasible set. Assumption (H2) excludes degenerate solutions with $\boldsymbol{\theta} = \mathbf{0}$

or $\mathbf{a} = \mathbf{0}$, which are already ruled out by the non-triviality condition in Theorem 1. Let $\mathbf{W} = (\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}) \in \mathbb{R}^{N+2J}$ denote the concatenated parameter vector, and define $F(\mathbf{W}) \equiv \ell(\boldsymbol{\theta}\mathbf{a}^\top + \mathbf{1}\mathbf{b}^\top)$ as the objective viewed as a function of $\mathbf{W}$.

With assumptions in place, we present the main convergence result in Theorem 2, established under the Kurdyka–Łojasiewicz (KL) framework (Łojasiewicz, 1963; Attouch et al., 2013; Bolte et al., 2014). The complete proof is provided in the Appendices A.3.

Theorem 2 (Convergence of `cBMM`)**.** *Under Assumptions* (H1) *and* (H2)*, the sequence* $\{(\boldsymbol{\theta}^{(t)}, \mathbf{a}^{(t)}, \mathbf{b}^{(t)})\}$ *generated by Algorithm 1 satisfies:*

(i) (Summability) $\sum_{t=0}^{\infty} \|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\| < \infty$*;*

(ii) (Whole-sequence convergence) $\mathbf{W}^{(t)} \to \mathbf{W}^* = (\boldsymbol{\theta}^*, \mathbf{a}^*, \mathbf{b}^*)$ *as* $t \to \infty$*;*

(iii) (Stationarity) $\mathbf{W}^*$ *satisfies the KKT conditions:*

$$\nabla_{\boldsymbol{\theta}} \ell(\mathbf{X}^*) = \mathbf{0}, \quad \mathbf{a}^* \geq \mathbf{0}, \quad \nabla_{\mathbf{a}} \ell(\mathbf{X}^*) \geq \mathbf{0}, \quad \mathbf{a}^* \circ \nabla_{\mathbf{a}} \ell(\mathbf{X}^*) = \mathbf{0}.$$

Furthermore, when the KL exponent is $1/2$*, the convergence is* ***linear****, i.e., there exist constants* $C > 0$ *and* $\alpha \in (0, 1)$ *such that* $\|\mathbf{W}^{(t)} - \mathbf{W}^*\| \leq C\alpha^t$.

4 Experiments

We conduct extensive experiments using both synthetic data and real-world applications. On the one hand, simulation studies on synthetic data are used for (i) sensitivity analysis under varying problem parameters, including the temperature, the missing pattern and rate, and the sparsity level of the discrimination parameter; (ii) comparison of `cBMM` and average accuracy for model ability ranking under varying degrees of item heterogeneity. Detailed results are provided in Appendices A.6 and A.7, with key findings summarized below. On the other hand, we evaluate `cBMM` on real-world benchmark suites to assess scalability by comparing its runtime against baseline methods, and to assess interpretability through the recovered model ability, item difficulty and discrimination.

Baseline methods. We compare `cBMM` with several baselines. First, we consider conventional IRT methods with two implementations: (i) `mirt` in `R` (Chalmers, 2012), which employs the EM algorithm for parameter estimation, and (ii) `py-irt` in Python (Lalor & Rodriguez, 2023), which uses stochastic variational inference (Kingma & Welling, 2013) with mini-batch gradient updates. We also include generic optimization approaches for solving (3). Specifically, we consider Limited-memory BFGS with Box constraints (`L-BFGS-B`) (Byrd et al., 1995), a second-order quasi-Newton method with bounded constraints, and `Manopt` (Boumal et al., 2014), a framework for optimization on manifolds. Implementation details are provided in Appendices A.5.

Key findings in simulation studies. As shown in Appendices A.6 and A.7, simulation studies on synthetic data reveal several key findings: (i) `cBMM` demonstrates robust performance under varying temperatures, missing patterns and rates, and sparsity levels of the discrimination parameter $\mathbf{a}$, consistently achieving comparable or even superior estimation accuracy while running significantly faster (up to 200x) than competing methods; (ii) `py-irt` produces substantially lower estimation accuracy than the other methods, and is therefore excluded from the real-world benchmark application studies; (iii) `cBMM` accurately recovers ground-truth ability rankings even under substantial item-discrimination heterogeneity, whereas average accuracy degrades markedly.

Real-world benchmark suites. We consider multiple benchmark suites with varying numbers of LLMs (N) and benchmark items (J) to assess scalability and interpretability. In total, we examine seven benchmark suites. The first, MATH-500 (Lightman et al., 2024), contains $J = 500$ items and is evaluated across $N = 140$ LLMs[1]; we adopt the evaluation data collected by Xu et al. (2025), with full model lists and prompt details provided in the Appendices A.9. The remaining six benchmark suites are drawn from the Hugging Face Open LLM Leaderboard (Fourrier et al., 2024), which

[1]The MATH-500 benchmark suite comprises 140 model–prompt configurations covering 70 distinct LLMs, each evaluated in both zero-shot and one-shot settings. Detailed specifications are provided in Appendix A.9.

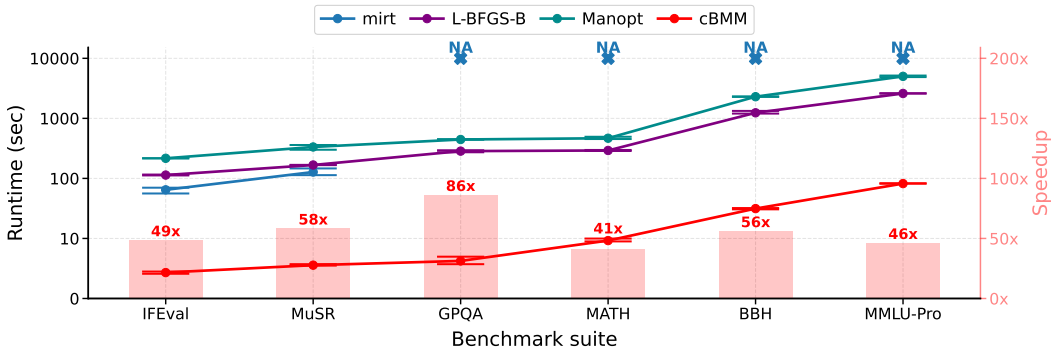


Figure 1: Scalability comparison across six large-scale benchmark suites. Runtime (sec) (left axis, points and interval markers) is shown on a logarithmic scale and reports the average runtime (points) as well as the minimum and maximum runtime (interval markers) across 50 replicates for each method. Speedup (right axis, shaded bars) represents the average speedup of `cBMM` over the fastest baseline. Benchmark suites are ordered by increasing scale of J with fixed $N = 2211$. Exact speedup ratios are annotated for each suite. 'NA' indicates convergence failure due to ill-conditioned data configurations in `mirt`.

includes $N = 2211$ LLMs evaluated between June and December 2024 under the Leaderboard v2 schema. These suites vary in the number of benchmark items (J): IFEval (541), MuSR (756), GPQA (1192), MATH (1324), BBH (5761), and MMLU-Pro (12032) (Zhou et al., 2023; Sprague et al., 2024; Rein et al., 2023; Hendrycks et al., 2021; Suzgun et al., 2023; Wang et al., 2024). We use the raw historical evaluation data obtained via the `huggingface_hub` API as curated by Wu et al. (2026), with further details available in the Appendices A.9.

4.1 Scalability

To demonstrate the scalability of `cBMM` compared to other baseline methods, we evaluate it on six large-scale benchmark suites from the Hugging Face Open LLM Leaderboard. For each suite, we conduct 50 replicates and report the average, minimum and maximum runtimes (in seconds) along with the average speedup ratio of `cBMM` relative to the fastest baseline.

Significant speedup and superior scalability. As illustrated in Figure 1, `cBMM` consistently outperforms existing baselines in computational efficiency, achieving a 41x-86x reduction in runtime. As the number of benchmark items J increases, `cBMM` exhibits improved scalability compared to generic optimization baselines. For instance, scaling from the IFEval suite ($J = 541$) to the MMLU-Pro suite ($J = 12032$) results in an average 508x increase in runtime for `Manopt`, a 392x increase in runtime for `L-BFGS-B`, but only a 104x increase for `cBMM`. Notably, `mirt` suffers from severe numerical instability and fails to converge (indicated by 'NA') on four large benchmark suites (GPQA, MATH, BBH, and MMLU-Pro), highlighting its limitations in large-scale settings. As a complement, our simulation studies in the Appendices A.6 further demonstrate that `cBMM` can achieve up to a 200x reduction in runtime while maintaining estimation accuracy comparable to, or even exceeding, that of baseline methods.

4.2 Interpretability of Model Ability

To interpret and validate the `cBMM`-recovered model abilities[2], we employ the MATH-500 benchmark suite, which encompasses a diverse set of LLMs released between July 2023 and September 2025 with varying model sizes. To facilitate comparison, we categorize these models into three scale tiers: small (0–7B), medium (7–20B), and large (>20B).

Consistency with parametric scaling laws. As shown in Figure 2, the `cBMM`-recovered model abilities $\hat{\boldsymbol{\theta}}$ exhibit a clear upward trend over release time across all three model scale tiers, con-

[2]For clarity, we omit the baseline methods in Sections 4.2–4.4, as they achieve recovery performance comparable to that of `cBMM` (see Appendix A.8).

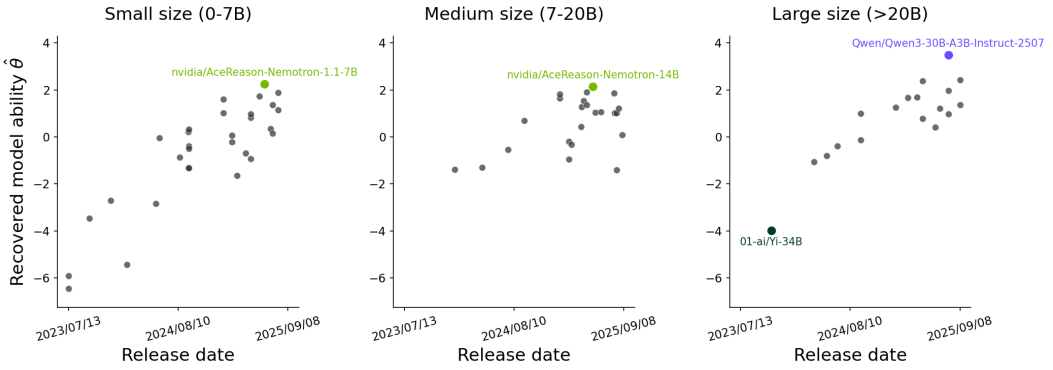


Figure 2: cBMM-recovered model abilities $\hat{\boldsymbol{\theta}}$ for LLMs in the MATH-500 benchmark suite. The three subplots (left, middle, right) correspond to three model scale tiers: small (0–7B), medium (7–20B), and large (>20B). All panels span the same release date range, from July 2023 to September 2025. LLMs with the highest $\hat{\boldsymbol{\theta}}$ are highlighted in each subplot, along with the LLM with the lowest $\hat{\boldsymbol{\theta}}$ in the large-size subplot.

sistent with empirical expectations. In addition, the parametric scaling effect, in line with Kaplan et al. (2020), becomes evident: larger models exhibit stronger reasoning capabilities, as reflected by higher overall placements and peak values within each tier. A notable deviation is observed for the 01-ai/Yi-34B model, which is primarily driven by the disproportionate scaling of training tokens relative to model parameters within the Yi series (Young et al., 2025). This phenomenon is consistent with the Chinchilla scaling laws (Hoffmann et al., 2022), which highlight the importance of balancing training data and model parameters for optimal model performance.

4.3 Interpretability of item difficulty

To interpret and validate the cBMM-recovered item difficulties, we use the human-annotated difficulty levels for each item in the MATH-500 benchmark suite as reference and assess the effectiveness of the estimated parameters $\hat{\mathbf{b}}$ obtained from cBMM.

Consistency with human-annotated difficulty. Figure 3, with subplots corresponding to overall and subject-wise results, displays the distribution of cBMM-recovered difficulty estimates $(-\hat{\mathbf{b}})$ across human-annotated difficulty levels (1–5) for the MATH-500 benchmark suite. A generally consistent upward trend is observed across all subplots at the median level, aligning well with the ordinal human-annotated difficulty scale. The 'Overall' subplot shows increasing variability in the distribution of $(-\hat{\mathbf{b}})$ as the difficulty level rises. We also observe several deviations from the overall increasing trend, such as level 1 in 'Geometry' and level 3 in 'Intermediate Algebra'. These discrepancies likely reflect a latent misalignment between model-perceived and human-judged difficulty, particularly for more challenging items. This observation is consistent with concerns raised by Eriksson et al. (2025) and Zhou et al. (2026), which suggest that benchmark datasets may lack sensitivity and specificity due to confounding features that influence LLM-perceived difficulty in ways not readily aligned with human intuition.

4.4 Interpretability of item discrimination

Regarding item discrimination, we hypothesize that greater heterogeneity in discrimination estimates leads to lower ranking consistency between the cBMM-recovered model abilities and average accuracies, since average accuracy implicitly assumes equal item discrimination. Specifically, we quantify ranking consistency using the Spearman correlation between the cBMM-recovered $\hat{\boldsymbol{\theta}}$ and average accuracies, a standard metric in IRT-based LLM evaluations (Castleman et al., 2025; Zhou et al., 2026). To measure heterogeneity in item discrimination, we use the coefficient of variation (CV) of the cBMM-recovered $\hat{\mathbf{a}}$, defined as the ratio of the standard deviation to the mean.

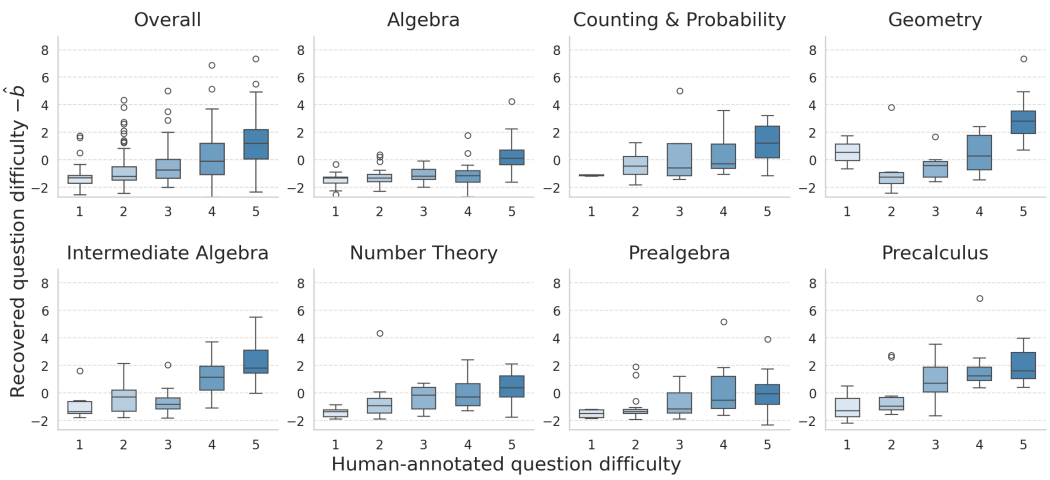


Figure 3: Distribution of cBMM-recovered item difficulties ($-\hat{\mathbf{b}}$) across human-annotated difficulty levels (1-5) in the MATH-500 benchmark suite, shown for the overall and subject-specific results.

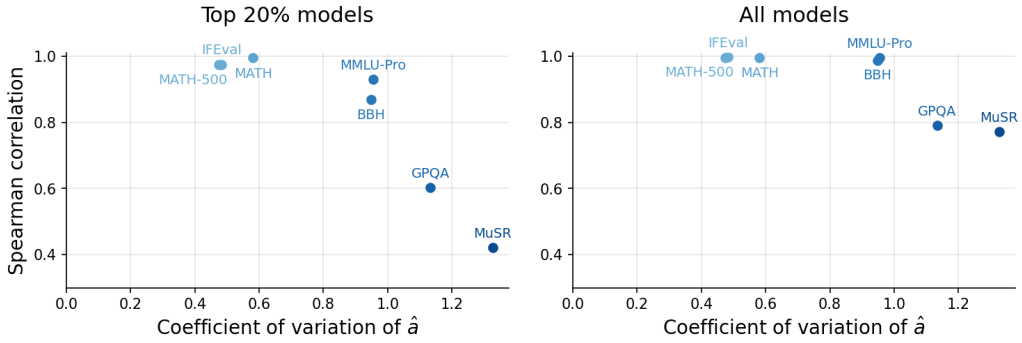


Figure 4: Spearman correlation between cBMM-recovered model abilities and average accuracies v.s. the coefficient of variation of cBMM-recovered $\hat{\mathbf{a}}$. The left panel shows results for the top 20% of models selected by average accuracy, while the right panel includes all LLMs. Each scatter point corresponds to a benchmark suite, as annotated in the plot.

Inverse relationship between ranking consistency and discrimination heterogeneity. Figure 4 depicts the relationship between the Spearman correlation of cBMM-recovered $\hat{\boldsymbol{\theta}}$ and average accuracies and the CV of cBMM-recovered $\hat{\mathbf{a}}$, across seven benchmark suites. To highlight different analysis scopes, we present results for the top 20% of models selected by average accuracy (left panel) and for the full set of LLMs (right panel). A clear inverse relationship is observed, providing empirical support for our hypothesis that greater heterogeneity in item discrimination estimates leads to more pronounced ranking discrepancies. A similar trend has also been reported in Castleman et al. (2025).

Fine-grained ranking shifts under discrimination heterogeneity. Figure 5 illustrates the detailed ranking flow between average accuracy and cBMM-recovered ability across three representative benchmark suites (MATH, MMLU-Pro, and GPQA). We select the top 30 models for MATH, top 50 for MMLU-Pro, and top 90 for GPQA, and visualize ranking transitions for LLMs that appear in both ranking schemes; full rankings are provided in the Supplementary Material. These ranking flows visually demonstrate that ranking discrepancies become increasingly pronounced as item discrimination grows more heterogeneous, offering a fine-grained view of ranking inconsistency. Additionally, we identify interesting patterns from the MATH benchmark: Qwen2.5-3B (Yang et al., 2025) and Ministral-8B-Instruct-2410 (Mistral AI, 2024) present significant upward shifts when transitioning from average accuracy to cBMM-recovered ability rakings, suggesting their capabilities may be underestimated by standard accuracy-based evaluations. Additional model representatives

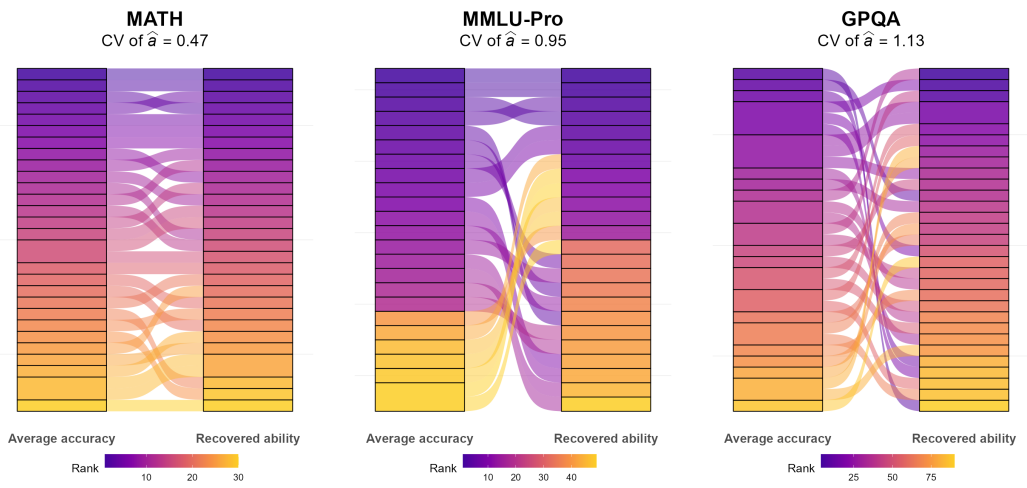


Figure 5: Ranking flow between average accuracy (left column in each panel) and cBMM-recovered ability (right column in each panel). The three subplots correspond to three benchmark suites: MATH, MMLU-Pro and GPQA, ordered by increasing coefficient of variation of the recovered $\hat{a}$. Only LLMs that consistently appear in the top tiers under both ranking schemes are shown; full rankings are provided in the Supplementary Material. Wider bars indicate tied rankings.

exhibiting significant rank shifts, together with uncertainty quantification for the estimated ranks, are provided in Appendix A.8.

5 Conclusion

This paper proposes an interpretable and scalable computational framework for evaluating LLMs, providing fine-grained assessments of model abilities and enabling large-scale real-world applications. Building on classical IRT and leveraging the MM principle, we reformulate the evaluation problem as a sequence of constrained matrix factorization problems and introduce the cBMM algorithm to solve it efficiently. We further establish theoretical guarantees for parameter identifiability and algorithmic convergence. Through extensive experiments on both synthetic data and large-scale real-world benchmarks, we demonstrate that cBMM achieves orders-of-magnitude computational speedups and improved scalability while maintaining comparable or higher estimation accuracy relative to existing methods. Moreover, its evaluation results are consistent with established parametric scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022) and human annotations, while revealing heterogeneity in model and item behavior that is not captured by average-accuracy-based evaluation.

Future work is motivated by a few current limitations. First, we treat model capability as a single dimensional trait, whereas general benchmarks, such as BBH and MMLU-Pro, may reveal multi-faceted model abilities. This motivates a natural extension to multidimensional IRT frameworks. We note that a corresponding implementation is already available in our GitHub repository, although additional theoretical challenges remain, particularly regarding identifiability under rotational invariance. Second, the logistic link in (1) can be extended to accommodate ordinal responses, such as those derived from LLM-as-a-Judge evaluations (Sheng et al., 2025). Third, beyond the bootstrap-based uncertainty quantification presented in Appendix A.8, establishing theoretical error bounds with finite-sample guarantees for the recovered parameters would provide additional rigor for both estimation accuracy and uncertainty.

AI use statement

In this work, we used generative AI tools to polish writing and retrieve related work. We have not used generative AI tools for research ideation or execution, draft sections, generating synthetic datasets, or proving mathematical claims. We have reviewed all AI-assisted work. For example, all AI-generated text refinements were manually verified to preserve the original scientific meaning and

logical intent, and all AI-assisted literature retrievals were thoroughly cross-checked for accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendices A.5 and A.9 provide detailed algorithmic and real-world benchmark specifications to facilitate reproducibility of our experimental results. The corresponding implementation is also available in our GitHub repository.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024. URL https://arxiv.org/abs/2303.08774.

Berk Atıl, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, et al. Non-determinism of 'deterministic' LLM system settings in hosted environments. In *Proceedings of the 5th Workshop on Evaluation and Comparison of NLP Systems*, pp. 135–148, Mumbai, India, December 2025. Association for Computational Linguistics. ISBN 979-8-89176-305-0. doi: 10.18653/v1/2025.eval4nlp-1.12. URL https://aclanthology.org/2025.eval4nlp-1.12/.

Hédy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1):91–129, 2013.

Frank B. Baker and Seock-Ho Kim (eds.). *Item Response Theory: Parameter Estimation Techniques*. CRC Press, 2nd edition, 2004. doi: 10.1201/9781482276725.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2026. URL https://arxiv.org/abs/2505.23281.

Sonia A. Bhaskar and Adel Javanmard. 1-bit matrix completion under exact low-rank constraint. In *2015 49th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, 2015. doi: 10.1109/CISS.2015.7086879.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014.

Nicolas Boumal, Bamdev Mishra, P.-A. Absil, and Rodolphe Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15(42):1455–1459, 2014. URL http://jmlr.org/papers/v15/boumal14a.html.

Ryan Burnell, Wout Schellaert, John Burden, Tomer D. Ullman, Fernando Martinez-Plumed, Joshua B. Tenenbaum, Danaja Rutar, Lucy G. Cheke, Jascha Sohl-Dickstein, Melanie Mitchell, et al. Rethink reporting of evaluation results in AI. *Science*, 380(6641):136–138, 2023. doi: 10.1126/science.adf6369. URL https://www.science.org/doi/abs/10.1126/science.adf6369.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. doi: 10.1137/0916069. URL https://doi.org/10.1137/0916069.

Tony Cai and Wen-Xin Zhou. A max-norm constrained minimization approach to 1-bit matrix completion. *Journal of Machine Learning Research*, 14(78):3619–3647, 2013. URL http://jmlr.org/papers/v14/cai13b.html.

Jane Castleman, Nimra Nadeem, Tanvi Namjoshi, and Lydia Liu. Rethinking math benchmarks: Implications for AI in education. In *Proceedings of the Innovation and Responsibility in AI-Supported Education Workshop*, volume 273 of *Proceedings of Machine Learning Research*, pp. 66–82. PMLR, 03 Mar 2025. URL https://proceedings.mlr.press/v273/castleman25a.html.

Jianzhe Chai, Yu Zhe, and Jun Sakuma. When benchmarks leak: Inference-time decontamination for LLMs. *arXiv preprint arXiv:2601.19334*, 2026. URL https://arxiv.org/abs/2601.19334.

R. Philip Chalmers. mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software*, 48(6):1–29, 2012. doi: 10.18637/jss.v048.i06. URL https://www.jstatsoft.org/index.php/jss/article/view/v048i06.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. URL https://arxiv.org/abs/2002.05709.

Yunxiao Chen, Xiaoou Li, Jingchen Liu, and Zhiliang Ying. Item response theory—a statistical framework for educational and psychological measurement. *Statistical Science*, 40(2):167 – 194, 2025. doi: 10.1214/23-STS896. URL https://doi.org/10.1214/23-STS896.

Zhiyu Zoey Chen, Jing Ma, Xinlu Zhang, Nan Hao, An Yan, Armineh Nourbakhsh, Xianjun Yang, Julian McAuley, Linda Petzold, and William Yang Wang. A survey on large language models for critical societal domains: Finance, healthcare, and law. *arXiv preprint arXiv:2405.01769*, 2024. URL https://arxiv.org/abs/2405.01769.

Neil Chowdhury, James Aung, Chan Jun Shern, Oliver Jaffe, Dane Sherburn, Giulio Starace, Evan Mays, Rachel Dias, Marwan Aljubeh, Mia Glaese, et al. Introducing SWE-bench verified, 2024. URL https://openai.com/index/introducing-swe-bench-verified/.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. URL https://arxiv.org/abs/2507.06261.

Mark A. Davenport, Yaniv Plan, Ewout van den Berg, and Mary Wootters. 1-bit matrix completion. *Information and Inference: A Journal of the IMA*, 3(3):189–223, 07 2014. ISSN 2049-8764. doi: 10.1093/imaiai/iau006. URL https://doi.org/10.1093/imaiai/iau006.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 9 1977. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1977.tb01600.x. URL https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

Yifei Dong, Fengyi Wu, Kunlin Zhang, Yilong Dai, Sanjian Zhang, Wanghao Ye, Sihan Chen, and Zhi-Qi Cheng. Large language model agents in finance: A survey bridging research, practice, and real-world deployment. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 17889–17907, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.972. URL https://aclanthology.org/2025.findings-emnlp.972/.

Maria Eriksson, Erasmo Purificato, Arman Noroozian, João Vinagre, Guillaume Chaslot, Emilia Gomez, and David Fernandez-Llorca. Can we trust AI benchmarks? an interdisciplinary review of current issues in AI evaluation. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(1):850–864, Oct. 2025. doi: 10.1609/aies.v8i1.36595. URL https://ojs.aaai.org/index.php/AIES/article/view/36595.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open LLM Leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Ronald K. Hambleton and Hariharan Swaminathan. *Item Response Theory: Principles and Applications*. Springer Dordrecht, Dordrecht, 1 edition, 1985. ISBN 978-0-89838-065-1. doi: 10.1007/978-94-017-1988-9. URL https://doi.org/10.1007/978-94-017-1988-9.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. URL https://arxiv.org/abs/2203.15556.

David R Hunter and Kenneth Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004. doi: 10.1198/0003130042836. URL https://doi.org/10.1198/0003130042836.

David Ilić and Gilles E. Gignac. Evidence of interrelated cognitive-like capabilities in large language models: Indications of artificial general intelligence or achievement? *Intelligence*, 106:101858, 2024. ISSN 0160-2896. doi: https://doi.org/10.1016/j.intell.2024.101858. URL https://www.sciencedirect.com/science/article/pii/S0160289624000527.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world Github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL https://arxiv.org/abs/2207.05221.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

Jonathan Kim, Anna Podlasek, Kie Shidara, Feng Liu, Ahmed Alaa, and Danilo Bernardo. Limitations of large language models in clinical problem-solving arising from inflexible reasoning. *Scientific Reports*, 15(1):39426, 2025. URL https://doi.org/10.1038/s41598-025-22940-0.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. URL https://arxiv.org/abs/1312.6114.

John Patrick Lalor and Pedro Rodriguez. py-irt: A scalable item response theory library for Python. *INFORMS Journal on Computing*, 35(1):5–13, 2023. doi: 10.1287/ijoc.2022.1250. URL https://doi.org/10.1287/ijoc.2022.1250.

Kenneth Lange, David R. Hunter, and Ilsoon Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000. doi: 10.1080/10618600.2000.10474858. URL https://doi.org/10.1080/10618600.2000.10474858.

Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, 1995. doi: 10.1137/1.9781611971217. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611971217.

Yucheng Li. Estimating contamination via perplexity: Quantifying memorisation in language model evaluation. *arXiv preprint arXiv:2309.10677*, 2023. URL https://arxiv.org/abs/2309.10677.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Anqi Lin, Junpu Ye, Chang Qi, Lingxuan Zhu, Weiming Mou, Wenyi Gan, Dongqiang Zeng, Bufu Tang, Mingjia Xiao, Guangdi Chu, et al. Bridging artificial intelligence and biological sciences: A comprehensive review of large language models in bioinformatics. *Briefings in Bioinformatics*, 26, 07 2025. URL https://doi.org/10.1093/bib/bbaf357.

Xiaoqian Liu, Xu Han, Eric C. Chi, and Boaz Nadler. A majorization-minimization Gauss-Newton method for 1-bit matrix completion. *Journal of Computational and Graphical Statistics*, 34(3): 1017–1029, 2025. doi: 10.1080/10618600.2024.2428610. URL https://doi.org/10.1080/10618600.2024.2428610.

Stanisław Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. *Les Équations aux Dérivées Partielles*, 117:87–89, 1963.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556/.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for general AI assistants. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 9025–9049, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/25ae35b5b1738d80f1f03a8713e405ec-Paper-Conference.pdf.

Mistral AI. Ministral-8B-Instruct-2410. https://huggingface.co/mistralai/Ministral-8B-Instruct-2410, 2024.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.441. URL https://aclanthology.org/2020.acl-main.441/.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

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. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, Mexico City, Mexico, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.301/.

Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv preprint arXiv:2310.10844*, 2023. URL https://arxiv.org/abs/2310.10844.

Huanxin Sheng, Xinyi Liu, Hangfeng He, Jieyu Zhao, and Jian Kang. Analyzing uncertainty of LLM-as-a-Judge: Interval evaluations with conformal prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11286–11328, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.569. URL https://aclanthology.org/2025.emnlp-main.569/.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. *arXiv preprint arXiv:2310.16049*, 2024. URL https://arxiv.org/abs/2310.16049.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Xiting Wang, Liming Jiang, Jose Hernandez-Orallo, David Stillwell, Luning Sun, Fang Luo, and Xing Xie. Evaluating general-purpose AI with psychometrics. *arXiv preprint arXiv:2310.16379*, 2023. URL https://arxiv.org/abs/2310.16379.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024. URL https://arxiv.org/abs/2406.01574.

Mike Wu, Richard L. Davis, Benjamin W. Domingue, Chris Piech, and Noah Goodman. Variational item response theory: Fast, accurate, and expressive. *arXiv preprint arXiv:2002.00276*, 2020. URL https://arxiv.org/abs/2002.00276.

Skyler Wu, Yash Nair, and Emmanuel J. Candès. Efficient evaluation of LLM performance with statistical guarantees. *arXiv preprint arXiv:2601.20251*, 2026. URL https://arxiv.org/abs/2601.20251.

Zhiyu Xu, Jia Liu, Yixin Wang, and Yuqi Gu. Latency-response theory model: Evaluating large language models via response accuracy and chain-of-thought length. *arXiv preprint arXiv:2512.07019*, 2025. URL https://arxiv.org/abs/2512.07019.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. Yi: Open foundation models by 01.AI. *arXiv preprint arXiv:2403.04652*, 2025. URL https://arxiv.org/abs/2403.04652.

Dingli Yu, Simran Kaur, Arushi Gupta, Jonah Brown-Cohen, Anirudh Goyal, and Sanjeev Arora. SKILL-MIX: A flexible and expandable family of evaluations for AI models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Jf5gplvglq.

Zicheng Zhang, Xiangyu Zhao, Xinyu Fang, Chunyi Li, Xiaohong Liu, Xiongkuo Min, Haodong Duan, Kai Chen, and Guangtao Zhai. Redundancy principles for MLLMs benchmarks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12492–12504, Vienna, Austria, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.acl-long.612/.

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. URL https://arxiv.org/abs/2311.07911.

Lexin Zhou, Lorenzo Pacchiardi, Fernando Martínez-Plumed, Katherine M. Collins, Yael Moros-Daval, Seraphina Zhang, Qinlin Zhao, Yitian Huang, Luning Sun, Jonathan E. Prunty, et al. General scales unlock AI evaluation with explanatory and predictive power. *Nature*, 652:58–67, Apr 2026. doi: 10.1038/s41586-026-10303-2. URL https://doi.org/10.1038/s41586-026-10303-2.

Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. PromptBench: A unified library for evaluation of large language models. *Journal of Machine Learning Research*, 25(254):1–22, 2024. URL http://jmlr.org/papers/v25/24-0023.html.

Yan Zhuang, Qi Liu, Zachary Pardos, Patrick C. Kyllonen, Jiyun Zu, Zhenya Huang, Shijin Wang, and Enhong Chen. Position: AI evaluation should learn from how we test humans. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 82483–82508. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhuang25e.html.

A APPENDICES

A.1 RELATED WORK

1-bit matrix completion. The optimization problem underlying LLM evaluation based on IRT can be viewed as a constrained 1-bit matrix completion problem with a user-specified rank. The 1-bit matrix completion problem has been well studied in the literature (Cai & Zhou, 2013; Davenport et al., 2014; Bhaskar & Javanmard, 2015; Liu et al., 2025), yet its computation remains challenging. `cBMM` builds on the MMGN framework (Liu et al., 2025), a recently proposed efficient algorithm for 1-bit matrix completion based on the MM principle. `cBMM` is specifically designed for the constrained formulation arising in LLM evaluation, and reformulates the original problem as a sequence of constrained matrix factorization subproblems. This leads to closed-form updates and a simple implementation based on standard linear algebra operations. As a result, `cBMM` is substantially faster than both IRT-based approaches and generic optimization methods, including quasi-Newton and manifold optimization algorithms.

IRT-based LLM evaluation. Conventional LLM evaluation relies on accuracy-based benchmarking in an aggregated manner (Chowdhury et al., 2024; Mialon et al., 2024), ignoring both the stochastic nature of LLM outputs and the heterogeneity of benchmark items (Li, 2023; Chai et al., 2026). Recently, methods from psychometrics, particularly Item Response Theory (IRT), have been adapted for LLM evaluation (Wang et al., 2023; Ilić & Gignac, 2024; Zhuang et al., 2025; Castleman et al., 2025), using a latent probabilistic framework to model LLM ability and item characteristics. Existing studies typically build on the two-parameter IRT model, whose conventional implementations often suffer from numerical instability and computational inefficiency (Chalmers, 2012; Xu et al., 2025), thereby limiting their applicability to small-scale settings. `cBMM` addresses these challenges by introducing a scalable and interpretable computational framework based on MM optimization. By ensuring both computational efficiency and stable parameter recovery, `cBMM` is well-suited for large-scale LLM evaluations.

A.2 PROOF OF THEOREM 1

Suppose a triplet $(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b})$ satisfies $\boldsymbol{\theta} \neq \mathbf{c}$ for some constant vector $\mathbf{c}$, and that $\mathbf{a}$ is nonnegative with at least one nonzero entry (without loss of generality, assume $a_1 > 0$). Consider a given score matrix $\mathbf{X} = \boldsymbol{\theta}\mathbf{a}^\top + \mathbf{1}\mathbf{b}^\top$. First, suppose that there exists a column $\mathbf{X}_{,j}$ that is neither a scalar multiple of the first column $\mathbf{X}_{,1}$ nor the vector of ones. Without loss of generality, suppose that this column is $\mathbf{X}_{,2}$. We argue that for any i, j, k,

$$\frac{x_{i1} - x_{j1}}{x_{j1} - x_{k1}} = \frac{x_{i2} - x_{j2}}{x_{j2} - x_{k2}}.$$

That is, the scaled differences among entries must be consistent. This is because the expression reduces to $\frac{(\theta_i - \theta_j)a_1}{(\theta_j - \theta_k)a_1} = \frac{(\theta_i - \theta_j)a_2}{(\theta_j - \theta_k)a_2}$. We can now identify $\boldsymbol{\theta}$ up to a scaling constant by first computing $\alpha \equiv \frac{a_1}{a_2} = \frac{x_{i1} - x_{j1}}{x_{i2} - x_{j2}}$ where x_{i1} and x_{j1} are distinct, which is guaranteed under the assumptions. We then solve

$$\mathbf{X}_{,1} - b_1\mathbf{1} = \alpha(\mathbf{X}_{,2} - b_2\mathbf{1})$$

to find b_1 and b_2. This further allows us to obtain $a_1\boldsymbol{\theta}$ through $a_1\boldsymbol{\theta} = \mathbf{X}_{,1} - b_1\mathbf{1}$. Since $a_1 > 0$, the rank ordering of $\boldsymbol{\theta}$ is uniquely determined.

Now consider any two non-identical rows i and j of $\mathbf{X}$. We have $\mathbf{X}_{i,} - \mathbf{X}_{j,} = (\theta_i - \theta_j)\mathbf{a}$. Hence, $\mathbf{a}$ can be identified up to a scaling constant through

$$\frac{a_k}{a_1} = \frac{x_{ik} - x_{jk}}{x_{i1} - x_{j1}}.$$

In particular, the rank ordering of $\mathbf{a}$ is determined. Finally, because $\boldsymbol{\theta}\mathbf{a}^\top$ is uniquely determined, $\mathbf{b}$ is also uniquely determined.

Now consider the case that every column aside from $\mathbf{X}_{,1}$ is either a scalar multiple of $\mathbf{X}_{,1}$ or the vector of ones. First note that any column i which is a scalar multiple of the vector of ones identifies a column where a_i is zero, and the corresponding entry of $\mathbf{b}$ is given by the value in that column. For

any column that is a scalar multiple of $\mathbf{X}_{,1}$, the previous arguments still hold to uniquely identify the rank ordering for $\mathbf{a}$ and $\boldsymbol{\theta}$, except that $\boldsymbol{b}$ is no longer uniquely determined, but rather follows the same ratios as the corresponding columns. Nonetheless, the rank ordering of $(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b})$ is uniquely determined.

A.3 Proof of Theorem 2

We begin by establishing a sufficient descent condition, which is the key step for applying the Kurdyka–Łojasiewicz (KL) framework (Łojasiewicz, 1963; Attouch et al., 2013; Bolte et al., 2014).

Lemma 1 (Sufficient Descent)**.** *Under Assumptions* (H1) *and* (H2)*, there exists a constant* $\rho > 0$ *such that for all* $t \geq 0$*,*

$$F(\mathbf{W}^{(t)}) - F(\mathbf{W}^{(t+1)}) \geq \rho \|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|^2.$$

Proof. Fix iteration t. Let $\tilde{\mathbf{X}}^{(t)} = \boldsymbol{\theta}^{(t)}(\mathbf{a}^{(t)})^\top + \mathbf{1}(\mathbf{b}^{(t)})^\top$ and $\tilde{\mathbf{Y}}^{(t)} = \tilde{\mathbf{X}}^{(t)} + 4\sigma \mathbf{Y} \circ \Phi(-\mathbf{Y} \circ \tilde{\mathbf{X}}^{(t)})$ as defined in Algorithm 1. Define the surrogate function at step t as $G(\mathbf{W} \mid \mathbf{W}^{(t)}) \equiv g(\boldsymbol{\theta}\mathbf{a}^\top + \mathbf{1}\mathbf{b}^\top \mid \tilde{\mathbf{X}}^{(t)})$.

We analyze each block update in Algorithm 1.

Step 1 (Update $\mathbf{a}$*, line 7).* With $\boldsymbol{\theta}^{(t)}$ and $\mathbf{b}^{(t)}$ fixed, the sub-objective reduces to $\sum_j \|\theta^{(t)} a_j - R_{1,\cdot,j}\|^2$, which is strongly convex in $\mathbf{a}$ with modulus $\mu_{\mathbf{a}} = \|\boldsymbol{\theta}^{(t)}\|^2 \geq c^2$ by Assumption (H2). By the strong convexity optimality inequality,

$$G(\mathbf{a}^{(t)} \mid \mathbf{W}^{(t)}) - G(\mathbf{a}^{(t+1)} \mid \mathbf{W}^{(t)}) \geq c^2 \|\mathbf{a}^{(t+1)} - \mathbf{a}^{(t)}\|^2. \tag{A.1}$$

Step 2 (Update $\mathbf{b}$*, line 8).* With $\boldsymbol{\theta}^{(t)}$ and $\mathbf{a}^{(t+1)}$ fixed, the sub-objective is strongly convex in $\mathbf{b}$ with modulus $\mu_{\mathbf{b}} = N$. Hence,

$$G(\mathbf{b}^{(t)} \mid \mathbf{W}^{(t)}) - G(\mathbf{b}^{(t+1)} \mid \mathbf{W}^{(t)}) \geq N \|\mathbf{b}^{(t+1)} - \mathbf{b}^{(t)}\|^2. \tag{A.2}$$

Step 3 (Update $\boldsymbol{\theta}$*, line 9).* With $\mathbf{a}^{(t+1)}$ and $\mathbf{b}^{(t+1)}$ fixed, the sub-objective is strongly convex in $\boldsymbol{\theta}$ with modulus $\mu_{\boldsymbol{\theta}} = \|\mathbf{a}^{(t+1)}\|^2 \geq c^2$ by Assumption (H2). Hence,

$$G(\boldsymbol{\theta}^{(t)} \mid \mathbf{W}^{(t)}) - G(\boldsymbol{\theta}^{(t+1)} \mid \mathbf{W}^{(t)}) \geq c^2 \|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\|^2. \tag{A.3}$$

Summing equation A.1–equation A.3 and letting $\rho_0 = \min(c^2, N) > 0$,

$$G(\mathbf{W}^{(t)} \mid \mathbf{W}^{(t)}) - G(\mathbf{W}^{(t+1)} \mid \mathbf{W}^{(t)}) \geq \rho_0 \|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|^2.$$

Applying the tangency condition $G(\mathbf{W}^{(t)} \mid \mathbf{W}^{(t)}) = F(\mathbf{W}^{(t)})$ and the domination condition $G(\mathbf{W}^{(t+1)} \mid \mathbf{W}^{(t)}) \geq F(\mathbf{W}^{(t+1)})$ yields the result with $\rho = \rho_0$. □

Lemma 2 (Relative Error Condition)**.** *Under Assumptions* (H1) *and* (H2)*, there exists a constant* $\lambda > 0$ *such that for all* $t \geq 0$*, there exists* $\mathbf{s}^{(t+1)} \in \partial F(\mathbf{W}^{(t+1)})$ *satisfying*

$$\|\mathbf{s}^{(t+1)}\| \leq \lambda \|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|.$$

Proof. We bound each component of the subgradient separately.

For $\boldsymbol{\theta}^{(t+1)}$ and $\mathbf{b}^{(t+1)}$, the corresponding block updates (lines 8–9 of Algorithm 1) are unconstrained least squares problems with exact closed-form solutions, so their first-order optimality conditions give $\nabla_{\boldsymbol{\theta}} G(\mathbf{W}^{(t+1)} \mid \mathbf{W}^{(t)}) = \mathbf{0}$ and $\nabla_{\mathbf{b}} G(\mathbf{W}^{(t+1)} \mid \mathbf{W}^{(t)}) = \mathbf{0}$. By the L-Lipschitz differentiability of ℓ and the double-linear bound $\|\mathbf{X}^{(t+1)} - \mathbf{X}^{(t)}\|_{\mathrm{F}} \leq M \|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|$ under Assumption (H1),

$$\|\nabla_{\boldsymbol{\theta}} F(\mathbf{W}^{(t+1)})\| \leq LM^2 \|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|,$$

and similarly for $\nabla_{\mathbf{b}} F(\mathbf{W}^{(t+1)})$.

For $\mathbf{a}^{(t+1)}$, the NNLS update (line 7) satisfies the KKT conditions: there exists $\boldsymbol{\mu} \geq \mathbf{0}$ with $\nabla_{\mathbf{a}} G(\mathbf{W}^{(t+1)} \mid \mathbf{W}^{(t)}) = \boldsymbol{\mu}$ and $\boldsymbol{\mu} \circ \mathbf{a}^{(t+1)} = \mathbf{0}$. Setting $\mathbf{s}_{\mathbf{a}}^{(t+1)} = \nabla_{\mathbf{a}} F(\mathbf{W}^{(t+1)}) - \boldsymbol{\mu} \in$

$\partial_{\mathbf{a}} F(\mathbf{W}^{(t+1)})$ and applying the same Lipschitz argument yields $\|\mathbf{s}_{\mathbf{a}}^{(t+1)}\| \leq LM^2\|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|$.

Combining all three components and setting $\lambda = 3LM^2$ completes the proof. □

With these two lemmas in place, we now restate the main convergence theorem and proceed to its full proof.

Theorem (Convergence of cBMM)**.** *Under Assumptions* (H1) *and* (H2)*, the sequence* $\{(\boldsymbol{\theta}^{(t)}, \mathbf{a}^{(t)}, \mathbf{b}^{(t)})\}$ *generated by Algorithm 1 satisfies:*

(i) (Summability) $\sum_{t=0}^{\infty} \|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\| < \infty$*;*

(ii) (Whole-sequence convergence) $\mathbf{W}^{(t)} \to \mathbf{W}^* = (\boldsymbol{\theta}^*, \mathbf{a}^*, \mathbf{b}^*)$ *as* $t \to \infty$*;*

(iii) (Stationarity) $\mathbf{W}^*$ *satisfies the KKT conditions:*

$$\nabla_{\boldsymbol{\theta}} \ell(\mathbf{X}^*) = \mathbf{0}, \quad \mathbf{a}^* \geq \mathbf{0}, \quad \nabla_{\mathbf{a}} \ell(\mathbf{X}^*) \geq \mathbf{0}, \quad \mathbf{a}^* \circ \nabla_{\mathbf{a}} \ell(\mathbf{X}^*) = \mathbf{0}.$$

Furthermore, when the KL exponent is $1/2$*, the convergence is* ***linear****, i.e., there exist constants* $C > 0$ *and* $\alpha \in (0, 1)$ *such that* $\|\mathbf{W}^{(t)} - \mathbf{W}^*\| \leq C\alpha^t$.

Proof. Step 1: Monotone decrease. By Lemma 1, $\{F(\mathbf{W}^{(t)})\}$ is monotonically non-increasing. Since $\ell(\mathbf{X}) \geq 0$ for all $\mathbf{X}$, the sequence converges to a finite limit $F^* \geq 0$.

Step 2: KL property. The function $F(\mathbf{W}) = \ell(\boldsymbol{\theta}\mathbf{a}^\top + \mathbf{1}\mathbf{b}^\top)$ is a composition of the logistic loss with a bilinear map. Since the logistic loss is sub-analytic and the bilinear map is polynomial, F is sub-analytic and hence belongs to an o-minimal structure (Attouch et al., 2013). Therefore F satisfies the KL property at every critical point, with KL exponent being $1/2$.

Step 3: Summability via KL inequality. By Assumption (H1), there exists a subsequence $\mathbf{W}^{(t_k)} \to \mathbf{W}^*$ with $F(\mathbf{W}^*) = F^*$. By the KL property, there exist $\eta > 0$, a neighborhood $\mathcal{U}$ of $\mathbf{W}^*$, and a concave function $\varphi : [0, \eta) \to \mathbb{R}_+$ with $\varphi(0) = 0$ and $\varphi' > 0$, such that for all $\mathbf{W} \in \mathcal{U}$ with $F^* < F(\mathbf{W}) < F^* + \eta$,

$$\varphi'(F(\mathbf{W}) - F^*)\|\mathbf{s}\| \geq 1, \quad \forall \mathbf{s} \in \partial F(\mathbf{W}).$$

For sufficiently large t, let $\Delta_t = \varphi(F(\mathbf{W}^{(t)}) - F^*) - \varphi(F(\mathbf{W}^{(t+1)}) - F^*)$. By concavity of φ, Lemma 1, the KL inequality, and Lemma 2,

$$\Delta_t \geq \varphi'(F(\mathbf{W}^{(t)}) - F^*) \cdot \rho\|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|^2 \geq \frac{\rho}{\lambda} \cdot \frac{\|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|^2}{\|\mathbf{W}^{(t)} - \mathbf{W}^{(t-1)}\|}.$$

Applying the AM-GM inequality $a^2/b \geq 2a - b$ and summing over $t = T, \ldots, T'$ yields a telescoping bound:

$$\sum_{t=T}^{T'} \|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\| \leq \frac{\lambda}{2\rho}\varphi(F(\mathbf{W}^{(T)}) - F^*) + \frac{1}{2}\|\mathbf{W}^{(T)} - \mathbf{W}^{(T-1)}\| < \infty.$$

Letting $T' \to \infty$ establishes (i).

Step 4: Whole-sequence convergence and stationarity. Part (i) implies $\{\mathbf{W}^{(t)}\}$ is a Cauchy sequence, so $\mathbf{W}^{(t)} \to \mathbf{W}^*$, establishing (ii). By Lemma 2, $\|\mathbf{s}^{(t+1)}\| \to 0$. Since ∂F has a closed graph, passing to the limit gives $\mathbf{0} \in \partial F(\mathbf{W}^*)$, which is equivalent to the KKT conditions in (iii). The linear convergence rate follows from the KL exponent of $1/2$ (Bolte et al., 2014). □

A.4 Per-iteration computational complexity

The computation bottleneck of Algorithm 1 lies in evaluating a matrix of the form $(\mathbf{u}\mathbf{v}_1^\top)_\Omega$, where $(\cdot)_\Omega$ is the masking operator that preserves entries $(i, j) \in \Omega$ and sets all others to zero. Note that the active set method (Lawson & Hanson, 1995) used to solve

$$\arg\min_{\mathbf{v}_1 \geq \mathbf{0}} \|\mathbf{u}\mathbf{v}_1^\top - \mathbf{R}\|_{\mathrm{F}(\Omega)}^2$$

in line 7 also requires computing a matrix of the form $(\mathbf{u}\mathbf{v}_1^\top)_\Omega$ at each of its iterations. Naively computing $\mathbf{u}\mathbf{v}_1^\top$ and then masking out entries in Ω^c costs $O(NJ)$ flops. However, this complexity can be reduced to $O(|\Omega|)$ by directly evaluating only the observed entries, which is a substantial improvement when $|\Omega| \ll NJ$. We detail this construction below.

Let $\mathbf{A} \equiv (\mathbf{u}\mathbf{v}_1^\top)_\Omega$ and let Ω_i denote the set of indices j such that $(i, j) \in \Omega$. Since we only need to compute entries $j \in \Omega_i$ for the i-th row of $\mathbf{A}$, we denote this nonzero subset as $\mathbf{A}(i, \Omega_i)$ and observe that

$$\mathbf{A}(i, \Omega_i) = u_i \mathbf{v}_{1,\Omega_i},$$

where $\mathbf{v}_{1,\Omega_i}$ is the sub-vector of $\mathbf{v}_1$ containing only the entries indexed by Ω_i. Consequently, computing the i-th row of $\mathbf{A}$ incurs a cost of $O(|\Omega_i|)$ flops. Summing over all N rows yields a total computational complexity of $O(|\Omega|)$ flops per-iteration.

A.5 Algorithmic details

EM algorithm for `mirt`. We implement by setting itemtype as 2PL in the `mirt` function of `R` (Chalmers, 2012). Furthermore, in the M-step, we specify `L-BFGS-B` optimizer (Byrd et al., 1995) embedded in the `R` function to ensure non-negativity constraint. Subsequently, Expected A Posteriori (EAP) method is employed to derive the final estimates of $\boldsymbol{\theta}$ post-convergence.

Post-hoc proximal gradient for `Manopt`. The optimization is carried out via conjugate gradient over the product space $(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}) \in \mathbb{R}^N \times \mathbb{R}_+^J \times \mathbb{R}^J$, and incorporates a post-hoc proximal gradient step to enforce the constraint $\mathbf{a} \geq 0$. Specifically, if denoting $(\hat{\boldsymbol{\theta}}, \hat{\mathbf{a}}, \hat{\mathbf{b}})$ as the optimum from vanilla `Manopt`, it projects

$$\begin{cases} \tilde{\mathbf{b}}^{(i+1)} = \tilde{\mathbf{b}}^{(i)} - \eta \nabla_{\tilde{\mathbf{b}}^{(i)}} \ell(\tilde{\mathbf{X}}^{(i)}) \\ \tilde{\mathbf{a}}^{(i+1)} = \text{proj}_{\mathbb{R}_+} \left(\tilde{\mathbf{a}}^{(i)} - \eta (\nabla_{\tilde{\mathbf{a}}^{(i)}} \ell(\tilde{\mathbf{X}}^{(i)}) + \lambda) \right) \end{cases}$$

with initial $(\tilde{\mathbf{a}}^{(1)}, \tilde{\mathbf{b}}^{(1)}) = (\hat{\mathbf{a}}, \hat{\mathbf{b}})$, where $\text{proj}_{\mathbb{R}_+}(x) = \max(0, x)$ is the proximal operator to non-negative orthant. η, λ control the step size and sparsity level. The final output then becomes $(\hat{\boldsymbol{\theta}}, \tilde{\mathbf{a}}, \tilde{\mathbf{b}})$.

Hyperparameters. In `mirt`, the latent model trait follows a prior distribution of $\theta_i \overset{\text{i.i.d.}}{\sim} \text{N}(0, 1)$, with initial values for item parameters $\log a_j \overset{\text{i.i.d.}}{\sim} \text{N}(0, 1)$ and $b_j \overset{\text{i.i.d.}}{\sim} \text{N}(0, 1)$. The same prior for $\boldsymbol{\theta}$ is carried over during EAP. Similarly, for `cBMM`, `L-BFGS-B` and `Manopt`, the related parameters θ_i, $\log a_j$, and b_j are initialized using the same distributions. All algorithms are set to a maximum of 1,000 iterations/epochs with a convergence tolerance of 10^{-4}. For `py-irt`, we adhere to default configurations as in the package except the epoch limit. Computations are executed on Intel Xeon E5-2683 v4 processors, where each replicate is assigned to one dedicated CPU core with 64 GB of allocated RAM.

A.6 Simulated studies for sensitivity analysis

We reexamine the computational scalability of different methods using synthetic examples, by varying the evaluation scale, defined by the number of models N and benchmark items J. Additionally, to demonstrate the robustness of `cBMM`, we conduct sensitivity analysis by varying several factors during synthetic data generation, including the temperature σ, the missing rate within $\mathbf{Y}$, $\rho = 1 - |\Omega|/NJ$ under different missingness patterns, and the redundancy level in benchmark items, as reflected by the sparsity level of $\mathbf{a}$, defined as $s = 1 - \|\mathbf{a}\|_0/J$. Here, $a_j = 0$ indicates that the j-th item lacks discriminative power for distinguishing model performance and is therefore considered redundant.

Data generation process. The default data generation process follows 50 replicates for each design. The response y_{ij} is generated under logit link (1) based on the underlying score x_{ij} reparameterized by (2), with $\theta_i \overset{\text{i.i.d.}}{\sim} \text{N}(0, 1)$ for $i = 1, ..., N$, $a_j \overset{\text{i.i.d.}}{\sim} \text{uniform}(0.5, 1)$ and $b_j \overset{\text{i.i.d.}}{\sim} \text{N}(0, 0.5)$ for $j = 1, ..., J$, which follows the same data generation procedure in Xu et al. (2025). Unless otherwise specified, default parameters are set to $\sigma = 1, \rho = 0, s = 0, N = 1000$, and $J = 1000$.

Performance metrics. We evaluate the performance of the compared methods using the following metrics. (i) The Spearman correlation coefficient between the estimate and the ground truth for $\boldsymbol{\theta}$ and $\mathbf{a}$. (ii) The Root Mean Square Error (RMSE) of the estimate for $\mathbf{b}$. (iii) The relative error of the score matrix, defined as $\|\mathbf{X} - \hat{\mathbf{X}}\|_{\mathrm{F}}^2 / \|\mathbf{X}\|_{\mathrm{F}}^2$. (iv) The Hellinger distance between the estimated and true distributions, $\Phi(\hat{\mathbf{X}})$ and $\Phi(\mathbf{X})$, which is defined as (Davenport et al., 2014; Cai & Zhou, 2013)

$$\frac{1}{NJ} \sum_{i,j} d_{\mathrm{H}}^2(\Phi(\hat{x}_{ij}), \Phi(x_{ij})),$$

where $d_{\mathrm{H}}^2(p, q) = (\sqrt{p} - \sqrt{q})^2 + (\sqrt{1-p} - \sqrt{1-q})^2$. The Hellinger distance is a standard metric between two distributions. It is non-negative and equals zero when the two distributions are identical, which provides a different perspective on estimation accuracy compared to the relative error. (v) Runtime in seconds.

A.6.1 Evaluation size

To account for variations in the scale of models and benchmark items across different leaderboards (Fourrier et al., 2024; Balunović et al., 2026), we reexamine the computational scalability of different methods by synthetic examples. We evaluate two scenarios: fixed N and fixed J. In both cases, we vary the other dimension over $\{200, 500, 1000, 2000, 5000\}$.

Figure A.1 and A.2 present results under fixed N and fixed J settings, respectively. Notably, we observe that `py-irt` exhibits substantially worse estimation performance than the other methods. This is evidenced by lower Spearman correlations between the estimated item discrimination parameters $\hat{\mathbf{a}}$ and their ground truth values, higher RMSE for the estimated item easiness parameters $\hat{\mathbf{b}}$, and higher relative errors in the recovered score matrix $\hat{\mathbf{X}}$, along with substantially longer runtimes under the same iteration budget. We therefore include `py-irt` for completeness only and do not regard it as a primary baseline for comparison.

We also observe algorithmic instability in `mirt`. When $J \in \{2000, 5000\}$, the `mirt` solver produces 'NA' results due to singularities in response patterns, a common issue in the psychometrics literature (Baker & Kim, 2004; Chalmers, 2012). This may occur, for instance, when items exhibit near-constant responses with minimal variance across respondents. Furthermore, when two items induce nearly identical response patterns, singularities can also arise when implementing `mirt`. Other factors, such as extreme temperatures, specific missing patterns, or item redundancy, further exacerbate these numerical instabilities, as demonstrated in Section A.6.2, A.6.3 and A.6.4. Overall, this highlights the limitations of `mirt` in large-scale and complex settings.

Regarding the recovery of the score matrix $\mathbf{X}$, both evaluation scale settings exhibit a consistent decrease in relative error as the number of models or benchmark questions increases. However, the recovery behaviors of $\boldsymbol{\theta}$, $\mathbf{a}$, and $\mathbf{b}$ differ across the two scenarios. Fixing N while increasing J leads to more accurate recovery of model abilities $\boldsymbol{\theta}$, but generally provides limited improvement in estimating item features $\mathbf{a}$ and $\mathbf{b}$, since only the evaluation coverage of each model is expanded through additional benchmark items. Conversely, fixing J while increasing N improves estimation of the item parameters, whereas the accuracy of model ability estimates remains relatively stable because the underlying benchmark set is unchanged. These findings are consistent with established observations in psychometric IRT literature (Hambleton & Swaminathan, 1985; Baker & Kim, 2004; Chen et al., 2025).

Combined with the runtime analysis, `cBMM` produces results that are highly consistent with empirical expectations while substantially reducing computational overhead, highlighting its strong scalability and practical applicability.

A.6.2 Temperature

The temperature σ acts as a pivotal scaling parameter for signal processing, with demonstrated efficacy in confidence calibration (Guo et al., 2017; Kadavath et al., 2022) and contrastive multimodal alignment (Chen et al., 2020; Radford et al., 2021). We therefore investigate the sensitivity of evaluation outcomes under different values of σ.

Figure A.3 illustrates performances of different methods across varying settings of σ where $\log_{10}(\sigma) \in \{-0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75\}$. We again observe algorithmic instability in

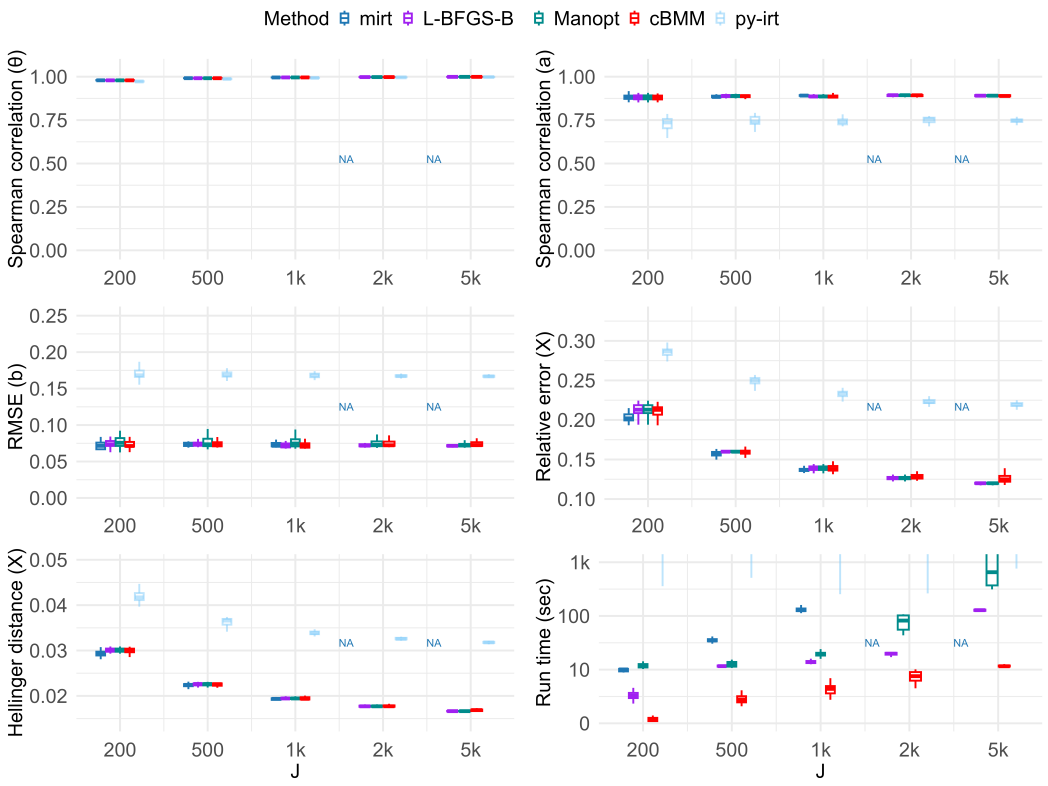


Figure A.1: Scalability comparison under varying numbers of items J. The number of models is fixed at $N = 1000$, while J increases from 200 to 5000. Each boxplot summarizes 50 independent replicates. Runtimes exceeding 1,000 seconds are truncated for better visualization. 'NA' indicates convergence failure due to ill-conditioned data configurations in `mirt`.

`mirt`: when $\log_{10}(\sigma) \in \{0.5, 0.75\}$, it produces 'NA' results due to singularities in response patterns, whereas `cBMM` remains numerically stable. Furthermore, `cBMM` demonstrates enhanced robustness across varying temperature levels. As evidenced in Figure A.3, in terms of RMSE for $\mathbf{b}$, relative error and Hellinger distance for $\mathbf{X}$, all methods exhibit performance degradation when the temperature is either too low or too high compared with moderate setting, which echoes the discussion in Davenport et al. (2014) regarding ill-posed recovery problem under extreme temperatures. Nevertheless, `cBMM` maintains consistently low variability across different values of σ, highlighting its robustness relative to competing methods. In contrast, the performance of `mirt` and `py-irt` follows a pronounced U-shaped pattern, with substantial fluctuations in the recovery of $\mathbf{b}$ and $\mathbf{X}$. Moreover, `py-irt` suffers from severe degradation in item discrimination estimation as temperature increases. These results further highlight the limitations of these methods in LLM evaluation tasks.

Compared with the two generic solvers `L-BFGS-B` and `Manopt`, `cBMM` exhibits comparable recovery performance while reducing runtime by one to two orders of magnitude (up to 200x speedup), depending on the temperature. This advantage is consistent across all temperature settings, underscoring the superior computational efficiency of `cBMM`.

A.6.3 Missingness

In this section, we analyze multiple missingness patterns: Missing Completely At Random (MCAR), Missing At Random (MAR) and Missing Not At Random (MNAR), with missing rates ($\rho = 1 - |\Omega|/(NJ)$). These patterns are designed to mimic real-world scenarios such as interfacing latency, generalization failures of LLMs on domain-specific items (Kim et al., 2025), or benchmark items triggering the safety protocols of participant models due to implicit unsafe flags (Shayegani et al., 2023).

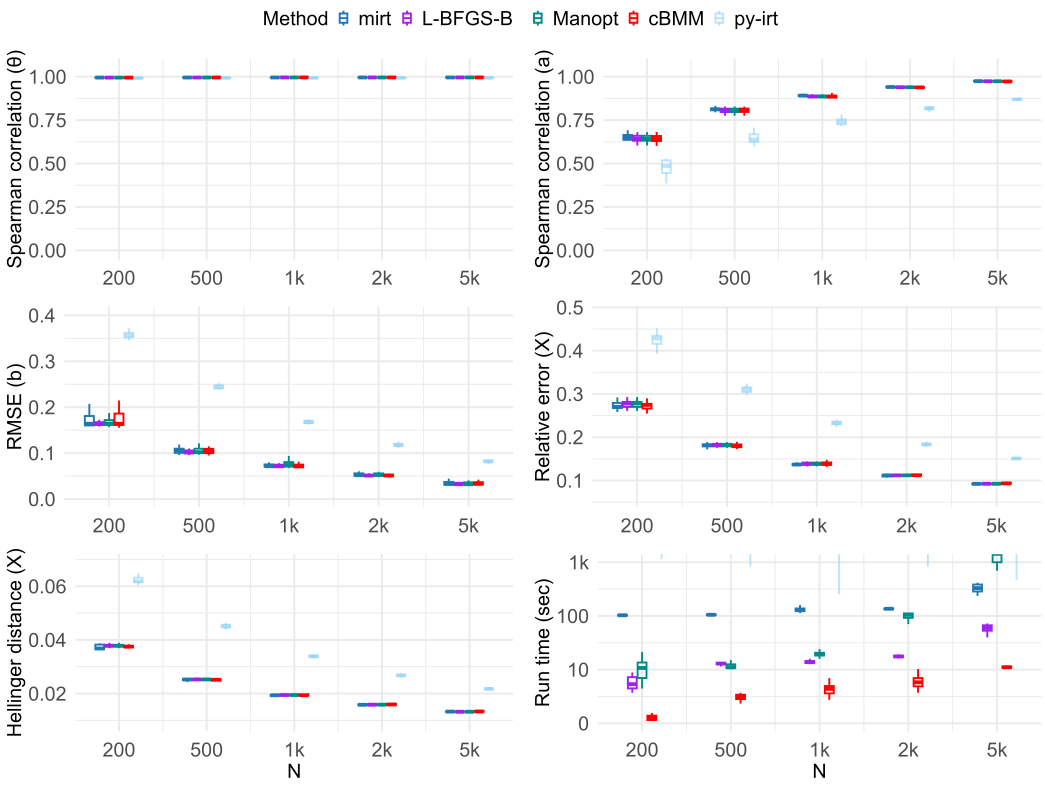


Figure A.2: Scalability comparison under varying numbers of LLMs N. The number of benchmark items is fixed at $J = 1000$, while N increases from 200 to 5000. Each boxplot summarizes 50 independent replicates. Runtimes exceeding 1,000 seconds are truncated for better visualization.

Missing Completely At Random. Interfacing latency, exceeded rate limits, or safety refusals often incur empty responses (Röttger et al., 2024; Xu et al., 2025), resulting in missing values in the response matrix $\mathbf{Y}$. Assuming an MCAR pattern that aligns with the external nature of these aforementioned failures, we vary the missing rate ρ from 0 to 0.8.

As illustrated in Figure A.4, `py-irt` consistently underperforms across nearly all recovered parameters, and is included as a baseline only for completeness. The other methods achieve comparable estimation accuracy across all metrics, and, as expected, their performance deteriorates as the missing rate increases. Notably, `cBMM` emerges as the most computationally efficient approach across all missing rates, achieving up to a 50x speedup over other methods.

Missing At Random. We design structural missing patterns based on the spatial positions of rows and columns to reflect potential generalization failures of LLMs on specific items. Specifically, we define a spatial metric Sp_{ij} as the average of the normalized row and column indices:

$$Sp_{ij} = \frac{r_i + c_j}{2}, \quad \text{where } r_i = \frac{i-1}{N-1},\ c_j = \frac{j-1}{J-1}.$$

The probability of entry (i, j) being missing is then determined by the logistic function

$$\Phi(\alpha + \beta \cdot Sp_{ij}) = \frac{1}{1 + \exp(-\alpha - \beta \cdot Sp_{ij})}$$

following equation (1), where $\beta = 2$ and α is calibrated to achieve a target overall missing rate ρ. This construction induces a graded missingness pattern, where the probability of missing entries increases with larger row or column indices.

The results are summarized in Figure A.5. We observe a pattern consistent with Figure A.4, where `cBMM` achieves comparable ranking recovery performance for $\boldsymbol{\theta}$ and $\mathbf{a}$, as well as comparable estimation accuracy for $\mathbf{b}$ and $\mathbf{X}$, while maintaining superior computational efficiency.

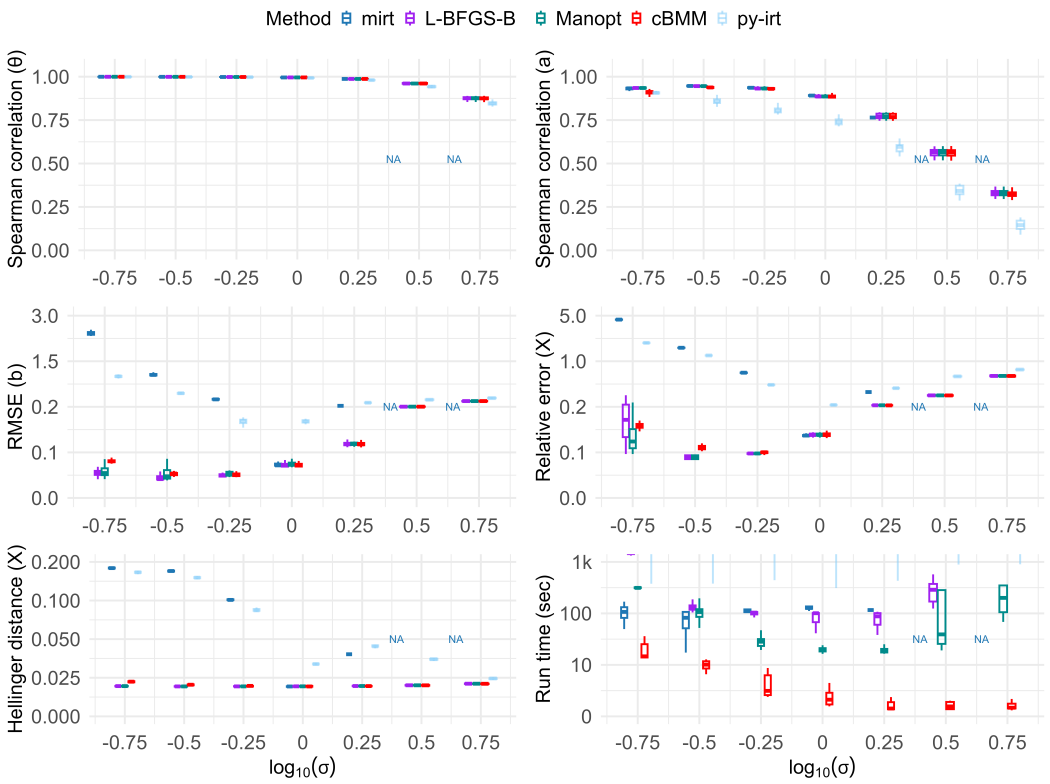


Figure A.3: Sensitivity analysis under varying temperature σ. Each boxplot summarizes 50 independent replicates. Runtimes exceeding 1,000 seconds are truncated for better visualization. 'NA' indicates convergence failure due to ill-conditioned data configurations in `mirt`.

Missing Not At Random. To simulate a MNAR pattern, we implement a masking mechanism for -1 entries (incorrect responses) in $\mathbf{Y}$. Specifically, we introduce a gate variable G_j for each column j, where $G_j \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\rho_c)$, and ρ_c represents the column-wise missing rate. If $G_j = 0$, no entries in column j are missing, while if $G_j = 1$, all -1 entries within that column are masked. This setting mimics scenarios where LLMs fail to generalize on domain-specific items or where benchmark items trigger safety mechanisms in participant models.

The results are summarized in Figure A.6. Notably, `mirt` consistently returns 'NA', even under less severe missingness patterns, due to numerical singularities induced by complex missing structures. In contrast, `cBMM` maintains high rank coherence of the recovered parameters and high estimation accuracy of latent scores, while achieving the shortest runtime among all methods.

A.6.4 Benchmark Redundancy

As benchmarks evolve, we expect to see increasing item redundancy across participant models (Zhang et al., 2025), which may stem from information leakage in training data over time (Chai et al., 2026). Such benchmark items provide limited discriminative power and should be phased out in future evaluations. A representative example is the evolution from SWE-bench (Jimenez et al., 2024) to SWE-bench Verified (Chowdhury et al., 2024), where a more discriminative subset of benchmark items was identified through extensive human annotation and verification. In contrast, we approach this problem from a data-driven perspective by identifying sparse discrimination parameters $\mathbf{a}$, thereby reducing reliance on manual curation and human labor.

We use the sparsity level of $\mathbf{a}$ to reflect the redundancy level in the benchmark and consider $s = 1 - \|\mathbf{a}\|_0/J \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ to validate method performance under varying levels of item redundancy, where the zero mask is assigned completely at random.

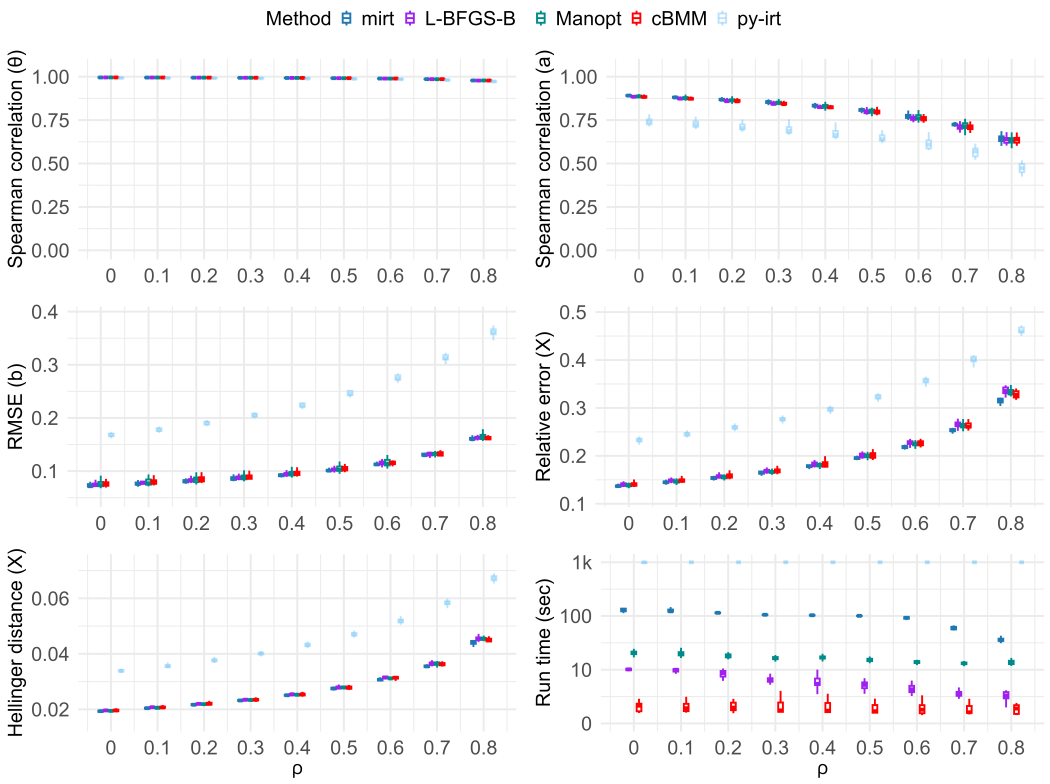


Figure A.4: Sensitivity analysis under varying missing rate ρ with an MCAR missingness pattern. Each boxplot summarizes 50 independent replicates. Runtimes exceeding 1,000 seconds are truncated for better visualization.

The results are summarized in Figure A.7. `py-irt` again exhibits significantly inferior recovery performance across all metrics. `mirt` reports 'NA' at $s \in \{0.4, 0.5\}$, as the singularity issue becomes more severe when benchmark items are increasingly redundant. In contrast, `cBMM` demonstrates consistently accurate parameter recovery while maintaining superior computational efficiency.

We further dissect the result by focusing on zero and non-zero patterns. In line with traditional hypothesis testing frameworks, we interpret the recovery of non-zero parameters as rejecting the null signal. Intuitively, $a_j = 0$ indicates that the j-th item has no discriminative power for distinguishing LLM abilities and should be identified and removed in subsequent iterations. We additionally consider two metrics to evaluate the accuracy of this discovery:

$$\text{Recall (a)} = \frac{TP}{TP + FN}, \quad \text{Precision (a)} = \frac{TP}{TP + FP},$$

where $TP = |\{j : \hat{a}_j \neq 0, a_j \neq 0\}|$, $FN = |\{j : \hat{a}_j = 0, a_j \neq 0\}|$ and $FP = |\{j : \hat{a}_j \neq 0, a_j = 0\}|$. Interestingly, all methods tend to over-reject the null: while this leads to a recall of 1, it fails to control the false discovery rate. This issue is particularly pronounced for `py-irt`, which fails to provide meaningful sparse recovery or redundancy identification. These results suggest that developing principled error-control mechanisms is an important direction for future work.

A.7 Advantages over Average Accuracy under Item Heterogeneity

We demonstrate that `cBMM` recovers model abilities more accurately than average accuracy under varying levels of item heterogeneity, measured by Spearman correlations with the ground-truth $\boldsymbol{\theta}$. Specifically, we vary the coefficient of variation (CV) of the item discrimination parameters while holding their mean fixed. The data-generating process follows the configuration described in Appendix A.6.

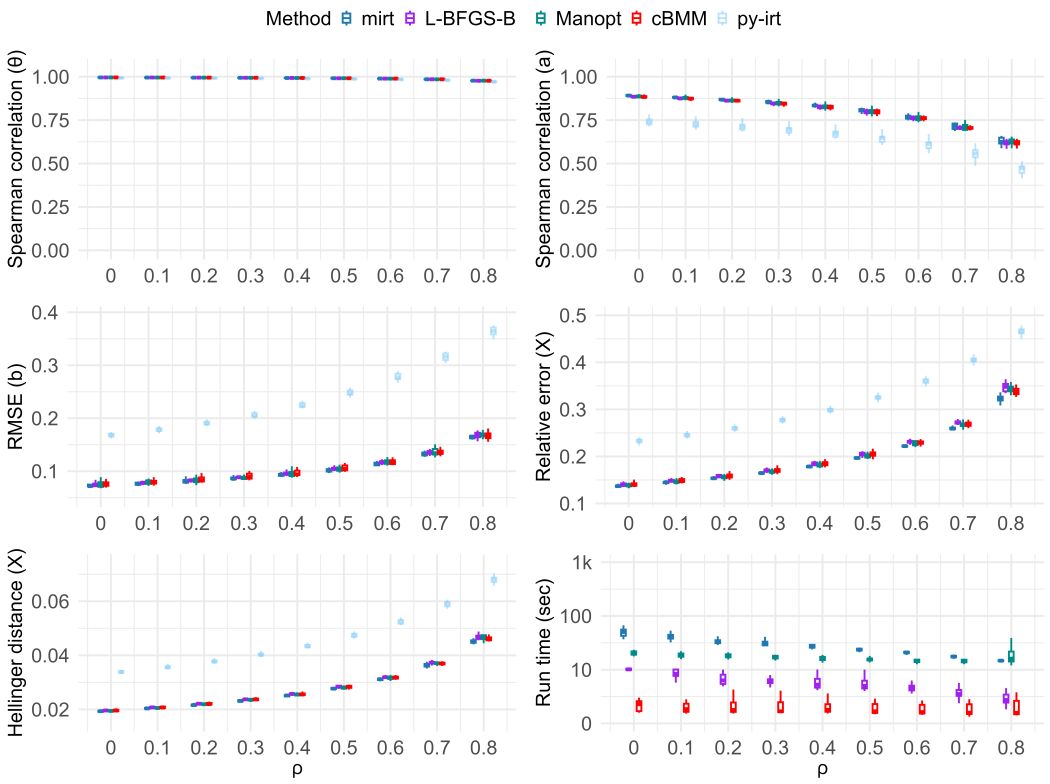


Figure A.5: Sensitivity analysis under varying missing rate ρ with an MAR missingness pattern. Each boxplot summarizes 50 independent replicates. Runtimes exceeding 1,000 seconds are truncated for better visualization.

As shown in Table A.1, `cBMM` and average accuracy perform similarly when item discrimination is relatively homogeneous (CV = 0.2). As item discrimination becomes more heterogeneous, however, `cBMM` maintains a high Spearman correlation with the true ability ranking, whereas the correlation for average accuracy steadily decreases. The advantage of `cBMM` becomes apparent at moderate heterogeneity and grows further in the high-CV settings, demonstrating more accurate recovery of latent ability when items differ substantially in discrimination.

Table A.1: Spearman correlation with the true model-ability ranking under increasing heterogeneityin item discrimination (mean $\pm$ standard deviation).

CV of $\mathbf{a}$	Spearman (`cBMM`-recovered $\hat{\boldsymbol{\theta}}$)	Spearman (average accuracy)
0.2	0.9642 ± 0.0035	0.9630 ± 0.0035
1.0	0.9778 ± 0.0017	0.9604 ± 0.0034
3.0	0.9867 ± 0.0024	0.9457 ± 0.0050
5.0	0.9867 ± 0.0020	0.9293 ± 0.0079
10.0	0.9848 ± 0.0022	0.8975 ± 0.0126
20.0	0.9822 ± 0.0026	0.8539 ± 0.0184
50.0	0.9770 ± 0.0031	0.7753 ± 0.0287

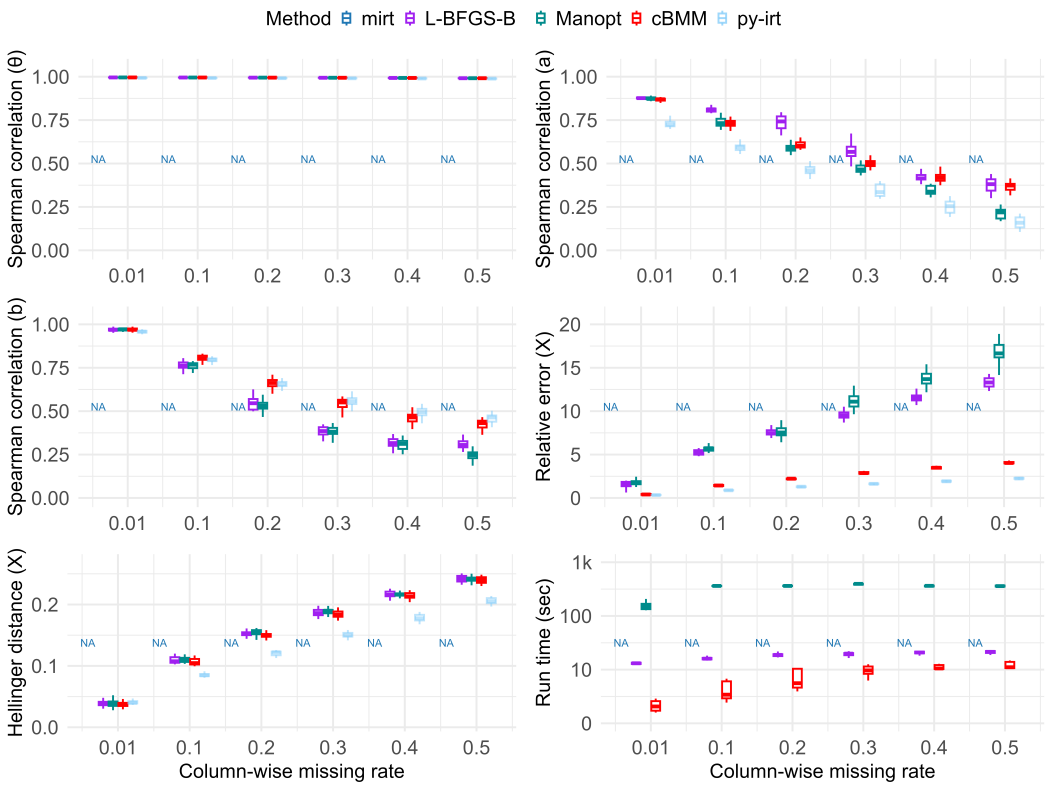


Figure A.6: Sensitivity analysis for missing rate ρ under MNAR pattern. Each boxplot summarizes 50 independent replicates. Runtimes exceeding 1,000 seconds are truncated for better visualization. 'NA' indicates convergence failure due to ill-conditioned data configurations in `mirt`.

A.8 Additional experiments on real-world benchmarks

A.8.1 Baseline agreement

Following Appendix A.6, we further examine the agreement between `cBMM` and the baseline methods `L-BFGS-B`, `Manopt`, and `py-irt` on real benchmark datasets. The results justify omitting these baselines from the downstream analyses in Sections 4.2, 4.3, and 4.4 of the main text.

We apply each method to MATH-500, GPQA, BBH, and MMLU-Pro and compute the Spearman correlation between the parameters estimated by `cBMM` and each baseline. `mirt` returned NA results and was therefore excluded. As shown in Table A.2, `cBMM` agrees almost perfectly with `L-BFGS-B` and `Manopt` for model ability $\hat{\boldsymbol{\theta}}$, with correlations ranging from 0.9992 to 0.9999, and remains highly consistent with `py-irt`, with correlations ranging from 0.9749 to 0.9981. The item parameters also show strong agreement between `cBMM` and `L-BFGS-B`/`Manopt`: correlations range from 0.9746 to 0.9971 for $\hat{\mathbf{a}}$ and from 0.9825 to 0.9985 for $\hat{\mathbf{b}}$. Agreement with `py-irt` is lower for item discrimination, consistent with the synthetic experiments in A.6. Recall that `py-irt` employs a stochastic gradient descent algorithm, which may lead to less accurate estimates than deterministic optimization methods.

Overall, the substantial speedup achieved by `cBMM` does not come at the expense of parameter recovery accuracy. Because the estimated model abilities are nearly identical across solvers, repeating the downstream analyses for each baseline would provide little additional information.

A.8.2 Uncertainty quantification in ability ranking

To quantify uncertainty in the estimated model-ability rankings, we perform an item-level bootstrap on the six benchmark suites in the Open LLM Leaderboard, stratifying by domain and category

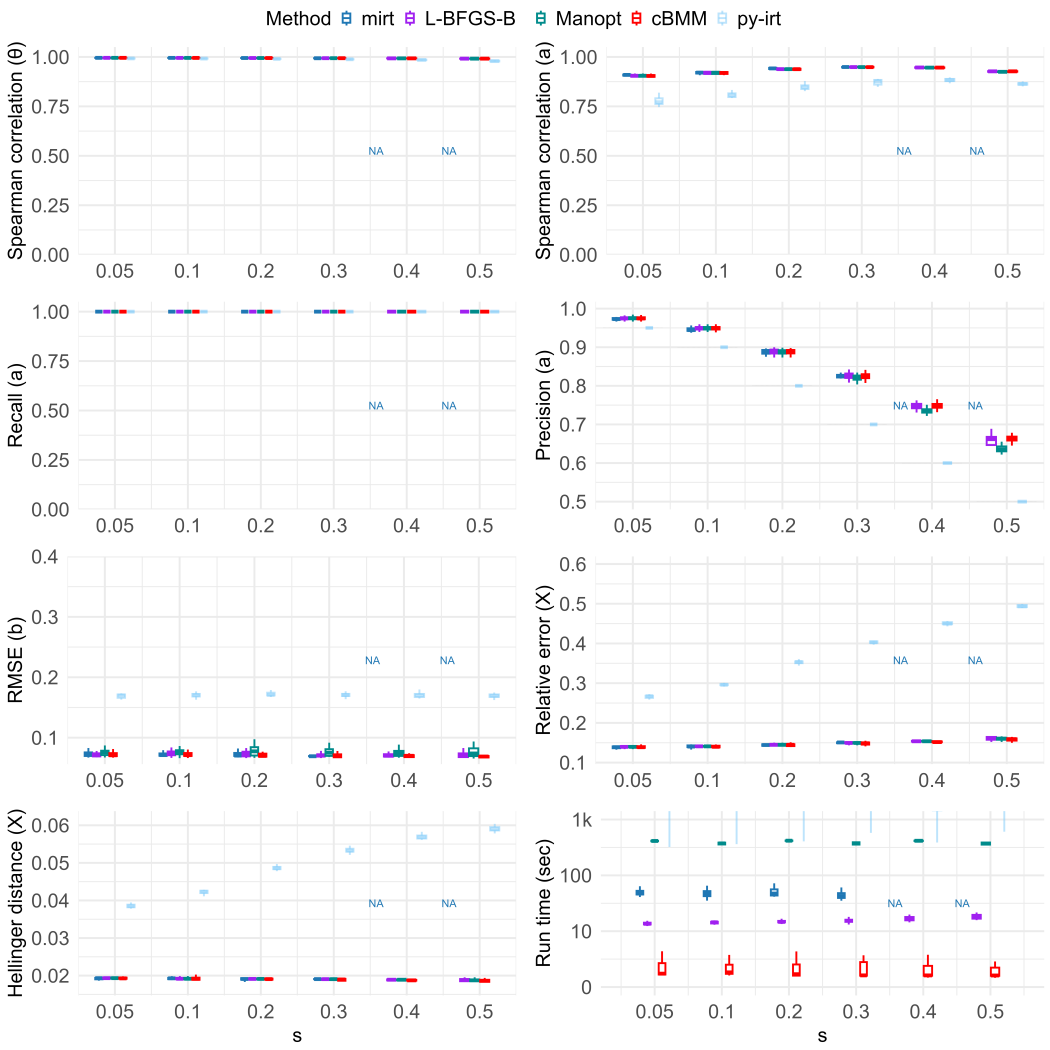


Figure A.7: Sparsity recovery analysis for benchmark redundancy detection. Each boxplot summarizes 50 independent replicates. Runtimes exceeding 1,000 seconds are truncated for better visualization. 'NA' indicates convergence failure due to ill-conditioned data configurations in `mirt`.

when available. For each of $B = 200$ bootstrap replicates, benchmark items are resampled with replacement within strata, `cBMM` is refitted, and model abilities and rankings are recomputed.

For each model, we construct a 90% bootstrap confidence interval (CI) for its rank using the 5th and 95th percentiles of the bootstrap rank distribution induced by the estimated $\hat{\boldsymbol{\theta}}$. We then evaluate the proportion of models exhibiting significant rank shifts, defined as cases where the standard-accuracy rank falls outside the corresponding `cBMM` 90% CI. As shown in Table A.3, the frequency of such rank shifts is strongly related to item heterogeneity, measured by the CV of $\hat{\mathbf{a}}$. More heterogeneous suites, such as MuSR and GPQA, exhibit substantially higher rates of significant rank shifts, reaching 85.62%, whereas more homogeneous suites, such as IFEval and MATH, exhibit lower rates of approximately 15%.

Tables A.4 and A.5 provide model-level results for IFEval and MATH. Even in these relatively homogeneous suites, the bootstrap analysis identifies notable rank shifts for several models. In IFEval, the NousResearch Hermes models and Meta's Llama-2-70b-chat-hf show upward shifts, whereas LGAI-EXAONE/EXAONE-3.5-2.4B-Instruct and arcee-ai/SuperNova-Medius show downward shifts. In MATH, Qwen/Qwen2.5-3B, ssmits/Qwen2.5-95B-Instruct, and various

Table A.2: Spearman correlation of estimates across method pairs on different benchmarks (mean $\pm$ standard deviation).

Method pair	MATH-500	GPQA	BBH	MMLU-Pro
Panel A: $\hat{\boldsymbol{\theta}}$ estimates				
`cBMM` vs `L-BFGS-B`	0.9997 ± 0.0000	0.9994 ± 0.0001	0.9992 ± 0.0000	0.9999 ± 0.0000
`cBMM` vs `Manopt`	0.9997 ± 0.0000	0.9994 ± 0.0001	0.9992 ± 0.0000	0.9999 ± 0.0000
`cBMM` vs `py-irt`	0.9962 ± 0.0002	0.9749 ± 0.0017	0.9962 ± 0.0002	0.9981 ± 0.0001
Panel B: $\hat{\mathbf{a}}$ estimates				
`cBMM` vs `L-BFGS-B`	0.9962 ± 0.0001	0.9782 ± 0.0010	0.9962 ± 0.0009	0.9971 ± 0.0001
`cBMM` vs `Manopt`	0.9922 ± 0.0001	0.9746 ± 0.0008	0.9922 ± 0.0009	0.9941 ± 0.0009
`cBMM` vs `py-irt`	0.8414 ± 0.0015	0.6369 ± 0.0073	0.8417 ± 0.0012	0.6726 ± 0.0007
Panel C: $\hat{\mathbf{b}}$ estimates				
`cBMM` vs `L-BFGS-B`	0.9985 ± 0.0002	0.9948 ± 0.0053	0.9985 ± 0.0002	0.9947 ± 0.0014
`cBMM` vs `Manopt`	0.9962 ± 0.0010	0.9299 ± 0.0111	0.9961 ± 0.0010	0.9825 ± 0.0021
`cBMM` vs `py-irt`	0.9953 ± 0.0002	0.8936 ± 0.0022	0.9953 ± 0.0002	0.9624 ± 0.0004

Table A.3: Proportion of models whose standard-accuracy rank falls outside the `cBMM` 90% bootstrap confidence interval (CI) for the rank. 'LB' and 'UB' denote the lower and upper bounds of the CI, respectively.

Benchmark suite	Below 90% CI LB	Above 90% CI UB	Outside 90% CI
IFEval (CV($\hat{\mathbf{a}}$)=0.48)	9.45%	6.29%	15.74%
MATH (CV($\hat{\mathbf{a}}$)=0.58)	6.60%	7.15%	13.75%
BBH (CV($\hat{\mathbf{a}}$)=0.95)	32.97%	36.59%	69.56%
MMLU-Pro (CV($\hat{\mathbf{a}}$)=0.95)	30.24%	34.09%	64.33%
GPQA (CV($\hat{\mathbf{a}}$)=1.13)	46.27%	39.35%	85.62%
MuSR (CV($\hat{\mathbf{a}}$)=1.33)	35.55%	34.01%	69.56%

fine-tuned Ministral variants shift upward, whereas a few earlier Qwen series and fine-tuned Llama 3 models shift downward.

A.8.3 Recovery stability

To validate the robustness of `cBMM` to initialization, we conduct experiments under various random initialization strategies following Castleman et al. (2025) on six real-world benchmark suites from the Open LLM Leaderboard. We run 50 replicates for each configuration to analyze the inter- and intra-run Spearman correlations. Specifically, for all $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, J\}$, we initialize $\theta_i \overset{\text{i.i.d.}}{\sim} \mathrm{N}(0, 1)$, $\log a_j \overset{\text{i.i.d.}}{\sim} \mathrm{N}(0, \sigma_a^2)$, and $b_j \overset{\text{i.i.d.}}{\sim} \mathrm{N}(0, \sigma_a^2)$, where $\sigma_a^2 \in \{1, 2, 3, 4, 5\}$, and set. We set a maximum of 1,000 iterations/epochs with a convergence tolerance of 10^{-5} when optimizing via `cBMM`. As shown in Table A.6, A.7, A.8, the recovered model and item parameters exhibit consistently high correlations across different seeds and initialization settings, suggesting that `cBMM` yields stable and reliable parameter recovery.

A.8.4 Warm-start for newly added models

A practical cold-start issue arises when newly added models must be evaluated. We therefore consider two complementary workflows for warm-starting the optimization when models are added sequentially.

Fixed-item scoring. If the benchmark and evaluation protocol remain unchanged, we can fix the previously calibrated item parameters and estimate only the new model's ability. This is a one-dimensional optimization problem, and a global refit is not required. To evaluate this workflow, we conducted leave-one-model-out replicates on each of the six Open LLM Leaderboard suites.

Table A.4: Selected models in the IFEval benchmark suite with rank shifts detected by the cBMM 90% CI.

Model representative	Average-accuracy rank	cBMM rank
NousResearch/Hermes-2-Theta-Llama-3-8B	405	372 (↑)
Qwen/Qwen2.5-Coder-7B-Instruct	537	468 (↑)
Qwen/Qwen2-VL-72B-Instruct	555	484 (↑)
NousResearch/Nous-Hermes-2-Mistral-7B-DPO	624	568 (↑)
meta-llama/Llama-2-70b-chat-hf	689	619 (↑)
NousResearch/Hermes-2-Pro-Llama-3-8B	719	636 (↑)
LGAI-EXAONE/EXAONE-3.5-2.4B-Instruct	65	153 (↓)
Qwen/Qwen2.5-Coder-32B-Instruct	255	349 (↓)
arcee-ai/SuperNova-Medius	296	364 (↓)

Table A.5: Selected models in the MATH benchmark suite with rank shifts detected by the cBMM 90% CI.

Model representative	Average-accuracy rank	cBMM rank
Qwen/Qwen2.5-3B	777	694 (↑)
prince-canuma/Ministral-8B-Instruct-2410-HF	924	818 (↑)
mistralai/Ministral-8B-Instruct-2410	938	827 (↑)
ssmits/Qwen2.5-95B-Instruct	1008	856 (↑)
Qwen/Qwen1.5-32B	136	188 (↓)
Qwen/Qwen1.5-14B	309	441 (↓)
grimjim/Llama-3-Nephilim-v2-8B	657	730 (↓)
grimjim/Llama-3-Instruct-8B-SPPO-Iter3-SimPO-merge	663	746 (↓)
grimjim/Llama-3-Nephilim-v3-8B	680	777 (↓)

According to Table A.9, relative to a full cold-start refit, fixed-item scoring reduced runtime by 83.94%–97.62%, while the resulting ability rankings had Spearman correlations of 0.9989–1.0000 with those from the full refit.

Periodic global recalibration. When a batch of new models has accumulated or the model-pool composition has changed substantially, cBMM can instead perform a joint global refit using the previous solution as a warm start. We evaluate this workflow by withholding a random batch equal to 1% of the original model pool and then reintroducing the batch. Compared with a randomly initialized global refit, as shown in Table A.10, warm-started recalibration reduced mean runtime by 29.12%–69.71% across the six suites, while producing ability rankings with Spearman correlations of 0.9971–1.0000.

A.9 Real-world benchmark details

MATH-500. We include the following list of LLM candidates in Table A.11 for the MATH-500 benchmark suite. Inference parameters, similar with Xu et al. (2025), are set as follows: temperature is 0.5, top-p is 0.95, and repetition penalty is 1.05, with a maximum output length of 10,240 tokens. The adopted prompt templates are shown in Table A.12, where {problem} represents the specific question for each item. Note that in MATH-500, $N = 140$ denotes 140 model-prompt configurations obtained from 70 distinct models, with each model evaluated under both zero-shot and one-shot prompting.

To assess whether jointly including both configurations affects our conclusions, we refitted cBMM separately using the 70 zero-shot configurations and the 70 one-shot configurations. We then compared each prompt-specific ability ranking with the ranking from the full MATH-500 fit restricted to the same 70 configurations. The Spearman correlations were 0.9996 ± 0.0001 (mean $\pm$ standard deviation) for zero-shot and 0.9997 ± 0.0001 for one-shot. Thus, the recovered rankings remain essentially unchanged when zero-shot and one-shot configurations are analyzed separately, indicating that our main MATH-500 conclusions are not driven by jointly including both prompting configurations.

Table A.6: Stability check for `cBMM`-recovered $\hat{\boldsymbol{\theta}}$. Each block corresponds to a specific benchmark suite. Cells report the mean and standard deviation (in parenthesis) of Spearman correlation coefficients across 50 replicates.

$\hat{\boldsymbol{\theta}}$	IFEval					BBH				
σ_a^2	1	2	3	4	5	1	2	3	4	5
1	1 (2.55e-17)					1 (5.70e-17)				
2	0.99 (9.58e-7)	1				0.99 (1.35e-7)	1 (5.09e-17)			
3	0.99 (2.07e-6)	1	1			0.99 (1.42e-7)	0.99 (3.02e-8)	1 (5.09e-17)		
4	0.99 (2.50e-6)	1	1	1		0.99 (1.47e-7)	0.99 (4.47e-8)	0.99 (1.71e-8)	1 (5.70e-17)	
5	0.99 (2.65e-6)	1	1	1	1	0.99 (1.49e-7)	0.99 (5.84e-8)	0.99 (3.30e-8)	0.99 (1.52e-8)	1 (7.20e-17)
	MATH					GPQA				
σ_a^2	1	2	3	4	5	1	2	3	4	5
1	1 (0.00e+0)					1 (2.55e-17)				
2	0.99 (1.75e-8)	1 (0.00e+0)				0.99 (1.90e-6)	1 (3.60e-17)			
3	0.99 (1.68e-8)	0.99 (1.91e-9)	1 (2.55e-17)			0.99 (2.56e-6)	0.99 (4.86e-7)	1 (3.60e-17)		
4	0.99 (1.67e-8)	0.99 (2.57e-9)	0.99 (9.11e-10)	1 (2.55e-17)		0.99 (2.83e-6)	0.99 (7.55e-7)	0.99 (9.82e-7)	1 (4.41e-17)	
5	0.99 (1.64e-8)	0.99 (3.12e-9)	0.99 (1.41e-9)	0.99 (8.90e-10)	1 (2.55e-17)	0.99 (2.88e-6)	0.99 (8.36e-7)	0.99 (8.06e-7)	0.99 (2.44e-7)	1 (4.41e-17)
	MuSR					MMLU-Pro				
σ_a^2	1	2	3	4	5	1	2	3	4	5
1	1 (4.85e-3)					1 (0.00e+0)				
2	1 (3.06e-5)	0.99 (5.00e-3)				0.99 (9.95e-8)	1 (7.85e-17)			
3	1 (7.84e-5)	1 (2.80e-5)	0.99 (4.87e-3)			0.99 (1.03e-7)	0.99 (1.26e-8)	1 (7.85e-17)		
4	1 (1.01e-4)	1 (4.34e-5)	1 (9.23e-6)	0.99 (4.87e-3)		0.99 (1.10e-7)	0.99 (1.52e-8)	0.99 (6.28e-9)	1 (7.85e-17)	
5	1 (1.13e-4)	1 (5.12e-5)	1 (8.37e-6)	1 (7.63e-6)	0.99 (4.92e-3)	0.99 (1.11e-7)	0.99 (1.77e-8)	0.99 (9.97e-9)	0.99 (3.66e-9)	1 (7.85e-17)

Hugging Face Open LLM Leaderboard. We refer to Wu et al. (2026) when formulating response matrix from Hugging Face's Open LLM Leaderboard. Specifically, we utilized the raw historical data M_1 from `https://github.com/skbwu/efficiently-evaluating-llms`, comprising item-level accuracy metadata for six major benchmarks: IFEval, BBH, MATH, GPQA, MuSR, and MMLU-Pro. The dataset was retrieved through the `huggingface_hub` API by iterating through a cohort of 2211 recent evaluation repositories, spanning the period from June to December 2024, under the Leaderboard v2 schema. Note that in February 2025, the Open LLM Leaderboard replaced its former SymPy-based MATH evaluator with Math-Verify and re-evaluated submitted models[3], leading to substantial rank changes, particularly for model families such as Qwen and DeepSeek. Consequently, the historical ranks in our analysis are not directly comparable with the model's position on the updated leaderboard.

Ranking flow details. Full ranking information based on average accuracy and recovered ability across the seven benchmarks is provided in the Supplementary Material (`.csv`).

[3]See `https://huggingface.co/blog/math_verify_leaderboard`.

Table A.7: Stability check for cBMM-recovered $\hat{\mathbf{a}}$. Each block corresponds to a specific benchmark suite. Cells report the mean and standard deviation (in parenthesis) of Spearman correlation coefficients across 50 replicates.

$\hat{\mathbf{a}}$	IFEval					BBH				
σ_a^2	1	2	3	4	5	1	2	3	4	5
1	1 (2.55e-17)					1 (1.14e-16)				
2	0.99 (9.58e-7)	1				0.99 (2.43e-5)	1 (1.14e-16)			
3	0.99 (2.07e-6)	1	1			0.99 (2.85e-5)	0.99 (6.16e-7)	1 (1.14e-16)		
4	0.99 (2.50e-6)	1	1	1		0.99 (2.61e-5)	0.99 (1.15e-6)	0.99 (3.18e-7)	1 (1.14e-16)	
5	0.99 (2.65e-6)	1	1	1	1	0.99 (2.18e-5)	0.99 (2.28e-6)	0.99 (1.30e-6)	0.99 (3.83e-7)	1 (1.14e-16)
	MATH					GPQA				
σ_a^2	1	2	3	4	5	1	2	3	4	5
1	1 (1.14e-16)					1 (0.00e+0)				
2	0.99 (3.63e-5)	1 (1.14e-16)				0.99 (9.34e-5)	1 (0.00e+0)			
3	0.99 (4.70e-5)	0.99 (8.51e-6)	1 (1.14e-16)			0.99 (7.47e-5)	0.99 (8.78e-5)	1 (0.00e+0)		
4	0.99 (4.95e-5)	0.99 (1.04e-5)	0.99 (1.13e-6)	1 (1.14e-16)		0.99 (1.17e-4)	0.99 (1.26e-4)	0.99 (9.80e-5)	1 (0.00e+0)	
5	0.99 (5.09e-5)	0.99 (1.08e-5)	0.99 (2.83e-6)	0.99 (8.30e-7)	1 (1.14e-16)	0.99 (1.18e-4)	0.99 (1.16e-4)	0.99 (1.01e-4)	0.99 (3.15e-5)	1 (0.00e+0)
	MuSR					MMLU-Pro				
σ_a^2	1	2	3	4	5	1	2	3	4	5
1	0.98 (1.40e-2)					1 (7.85e-17)				
2	1 (6.63e-4)	0.98 (1.45e-2)				0.99 (1.44e-5)	1 (5.55e-17)			
3	1 (9.82e-4)	1 (4.63e-4)	0.98 (1.44e-2)			0.99 (1.73e-5)	0.99 (3.19e-6)	1 (5.55e-17)		
4	1 (9.89e-4)	1 (6.22e-4)	1 (3.41e-4)	0.98 (1.42e-2)		0.99 (1.67e-5)	0.99 (1.40e-5)	0.99 (3.76e-6)	1 (0.00e+0)	
5	1 (1.10e-3)	1 (6.84e-4)	1 (3.46e-4)	1 (3.14e-4)	0.98 (1.44e-2)	0.99 (2.00e-5)	0.99 (2.57e-5)	0.99 (1.07e-5)	0.99 (1.76e-6)	1 (0.00e+0)

Table A.8: Stability check for cBMM-recovered $\hat{\mathbf{b}}$. Each block corresponds to a specific benchmark suite. Cells report the mean and standard deviation (in parenthesis) of Spearman correlation coefficients across 50 replicates.

$\hat{\mathbf{b}}$	IFEval					BBH				
σ_a^2	1	2	3	4	5	1	2	3	4	5
1	1 (2.55e-17)					1 (0.00e+0)				
2	0.99 (9.58e-7)	1				0.99 (8.77e-7)	1 (0.00e+0)			
3	0.99 (2.07e-6)	1	1			0.99 (9.51e-7)	0.99 (1.00e-7)	1 (8.82e-17)		
4	0.99 (2.50e-6)	1	1	1		0.99 (1.03e-6)	0.99 (3.07e-7)	0.99 (6.15e-8)	1 (1.02e-16)	
5	0.99 (2.65e-6)	1	1	1	1	0.99 (1.12e-6)	0.99 (5.10e-7)	0.99 (1.86e-7)	0.99 (4.91e-8)	1 (5.09e-17)
	MATH					GPQA				
σ_a^2	1	2	3	4	5	1	2	3	4	5
1	1 (0.00e+0)					1 (2.55e-17)				
2	0.99 (9.23e-5)	1 (0.00e+0)				0.99 (1.90e-6)	1 (2.55e-17)			
3	0.99 (1.38e-4)	0.99 (8.82e-5)	1 (0.00e+0)			0.99 (2.21e-6)	0.99 (1.12e-6)	1 (3.60e-17)		
4	0.99 (1.50e-4)	0.99 (1.16e-4)	0.99 (3.94e-5)	1 (0.00e+0)		0.99 (2.72e-6)	0.99 (1.78e-6)	0.99 (1.60e-6)	1 (5.70e-17)	
5	0.99 (1.51e-4)	0.99 (1.19e-4)	0.99 (4.28e-5)	0.99 (4.71e-6)	1 (8.82e-17)	0.99 (3.46e-6)	0.99 (2.40e-6)	0.99 (1.63e-6)	0.99 (3.26e-7)	1 (6.74e-17)
	MuSR					MMLU-Pro				
σ_a^2	1	2	3	4	5	1	2	3	4	5
1	1 (1.90e-3)					1 (5.55e-17)				
2	1 (1.18e-3)	1 (2.01e-3)				0.99 (1.95e-6)	1 (7.85e-17)			
3	1 (1.50e-3)	1 (5.09e-4)	1 (2.17e-3)			0.99 (2.78e-6)	0.99 (2.33e-7)	1 (7.85e-17)		
4	1 (1.51e-3)	1 (5.15e-4)	1 (2.85e-6)	1 (2.17e-3)		0.99 (3.05e-6)	0.99 (3.77e-7)	0.99 (5.36e-8)	1 (5.55e-17)	
5	1 (1.48e-3)	1 (5.42e-4)	1 (5.07e-6)	1 (6.26e-6)	1 (2.18e-3)	0.99 (3.20e-6)	0.99 (4.40e-7)	0.99 (1.21e-7)	0.99 (2.04e-8)	1 (0.00e+0)

Table A.9: Fixed-item scoring for one newly added model (mean $\pm$ standard deviation).

Benchmark suite	Runtime reduction (%)	Spearman correlation
IFEval	88.35 ± 0.82	$1.0000 \pm 2.8 \times 10^{-5}$
MuSR	87.53 ± 0.53	$0.9994 \pm 6.2 \times 10^{-4}$
GPQA	83.94 ± 2.66	$0.9996 \pm 1.9 \times 10^{-4}$
MATH	97.62 ± 1.73	$0.9989 \pm 1.2 \times 10^{-3}$
BBH	86.78 ± 0.46	$1.0000 \pm 3.2 \times 10^{-5}$
MMLU-Pro	90.28 ± 0.21	$1.0000 \pm 4.3 \times 10^{-5}$

Table A.10: Warm-started global recalibration after adding 1% of the original model pool (mean $\pm$ standard deviation).

Benchmark suite	Runtime reduction (%)	Spearman correlation
IFEval	63.93 ± 4.22	0.9999 ± 0.0000
MuSR	69.71 ± 6.32	0.9971 ± 0.0027
GPQA	59.57 ± 8.19	0.9990 ± 0.0003
MATH	64.20 ± 40.65	0.9981 ± 0.0004
BBH	55.06 ± 16.97	0.9998 ± 0.0002
MMLU-Pro	29.12 ± 18.11	1.0000 ± 0.0001

Table A.11: List of LLM candidates in the MATH-500 suite

- 01-ai/Yi-34B
- baidu/ERNIE-4.5-21B-A3B-PT
- baidu/ERNIE-4.5-21B-A3B-Thinking
- deepseek-ai/DeepSeek-R1-0528-Qwen3-8B
- deepseek-ai/DeepSeek-R1-Distill-Llama-8B
- deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B
- deepseek-ai/DeepSeek-R1-Distill-Qwen-14B
- deepseek-ai/DeepSeek-R1-Distill-Qwen-32B
- deepseek-ai/DeepSeek-R1-Distill-Qwen-7B
- dphn/dolphin-2.9.1-yi-1.5-34b
- dphn/Dolphin-Mistral-24B-Venice-Edition
- google/gemma-2-27b-it
- google/gemma-2b-it
- google/gemma-3-1b-pt
- HuggingFaceTB/SmolLM3-3B
- huihui-ai/Huihui-gpt-oss-20b-BF16-abliterated
- huihui-ai/Huihui-Qwen3-8B-abliterated-v2
- ibm-granite/granite-3.3-2b-instruct
- internlm/internlm2-chat-20b
- LGAI-EXAONE/EXAONE-4.0.1-32B
- LLM360/K2-Think
- meta-llama/Llama-2-7b-chat-hf
- meta-llama/Llama-2-7b-hf
- meta-llama/Llama-3.1-8B-Instruct
- meta-llama/Llama-3.2-1B-Instruct
- meta-llama/Llama-3.2-3B-Instruct
- meta-llama/Meta-Llama-3-8B-Instruct
- microsoft/Phi-3.5-mini-instruct
- microsoft/Phi-4-mini-instruct
- microsoft/Phi-4-mini-reasoning
- microsoft/Phi-4-mini-reasoning-plus
- mistralai/Mistral-7B-Instruct-v0.1
- mistralai/Mistral-7B-Instruct-v0.2
- mistralai/Mistral-7B-Instruct-v0.3
- mistralai/Mistral-Small-Instruct-2409
- moonshotai/Moonlight-16B-A3B
- moonshotai/Moonlight-16B-A3B-Instruct
- nvidia/AceReason-Nemotron-1.1-7B
- nvidia/AceReason-Nemotron-14B
- nvidia/Llama-3.1-Nemotron-8B-UltraLong-4M-Instruct
- nvidia/Nemotron-Research-Reasoning-Qwen-1.5B
- nvidia/NVIDIA-Nemotron-Nano-12B-v2
- nvidia/NVIDIA-Nemotron-Nano-9B-v2
- nvidia/OpenReasoning-Nemotron-1.5B
- nvidia/OpenReasoning-Nemotron-7B
- openai/gpt-oss-20b
- openbmb/MiniCPM4.1-8B
- Qwen/Qwen1.5-32B
- Qwen/Qwen2.5-0.5B-Instruct
- Qwen/Qwen2.5-1.5B-Instruct
- Qwen/Qwen2.5-14B-Instruct
- Qwen/Qwen2.5-32B-Instruct
- Qwen/Qwen2.5-3B-Instruct
- Qwen/Qwen2.5-7B-Instruct
- Qwen/Qwen2-7B-Instruct
- Qwen/Qwen3-0.6B
- Qwen/Qwen3-1.7B
- Qwen/Qwen3-4B-Thinking-2507
- Qwen/Qwen3-8B
- Qwen/QwQ-32B
- Qwen/Qwen3-14B
- Qwen/Qwen3-30B-A3B
- Qwen/Qwen3-30B-A3B-Instruct-2507
- Qwen/Qwen3-30B-A3B-Thinking-2507
- Qwen/Qwen3-32B
- Qwen/Qwen3-4B
- Qwen/Qwen3-4B-Instruct-2507
- swiss-ai/Apertus-8B-Instruct-2509
- zai-org/GLM-4-32B-0414
- zai-org/GLM-4-9B-0414

Table A.12: Prompt templates used in MATH-500 evaluation.

Zero-shot	**One-shot**
Solve the following math problem. Be clear and concise. Problem: "{problem}" Provide a **step-by-step solution**. Start each step with a number followed by a period (e.g., '1.', '2.', etc.). Use basic LaTeX for mathematical expressions, such as for fractions, exponents, and variables. Avoid complex formatting. At the very end of your entire response, and only at the very end, state the final answer. This final answer must be enclosed in a single LaTeX box, like so: $\boxed{\text{Answer}}$.	Solve the following math problem. Please think **step-by-step** to obtain the solution. Use basic LaTeX for mathematical expressions, such as for fractions, exponents, and variables. Avoid complex formatting. At the very end of your entire response, and only at the very end, state the final answer. This final answer must be enclosed in a single LaTeX box, like so: $\boxed{\text{Answer}}$. Here is an example of how to format your response and think about solving the problem: Example Problem: What is the sum of the two values of x for which $(x+3)^2 = 121$? Example Solution: Expanding the left side, we have $x^2 + 6x + 9 = 121 \Rightarrow x^2 + 6x - 112 = 0$. For a quadratic with the equation $ax^2 + bx + c = 0$, the sum of the roots is $-b/a$. Applying this formula to the problem, we have that the sum of the two roots is $-6/1 = \boxed{-6}$. Solution: $\boxed{-6}$ New Problem: {problem}.