RANKLLM: WEIGHTED RANKING OF LLMS BY QUANTIFYING QUESTION DIFFICULTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Benchmarks establish a standardized evaluation framework to systematically assess the performance of large language models (LLMs), facilitating objective comparisons and driving advancements in the field. However, existing benchmarks fail to differentiate question difficulty, limiting their ability to effectively distinguish models' capabilities. To address this limitation, we propose RankLLM, a novel framework designed to quantify both question difficulty and model competency. RankLLM introduces difficulty as the primary criterion for differentiation, enabling a more fine-grained evaluation of LLM capabilities. RankLLM's core mechanism facilitates bidirectional score propagation between models and questions. The core intuition of RankLLM is that a model earns a competency score when it correctly answers a question, while a question's difficulty score increases when it challenges a model. Using this framework, we evaluate 30 models on 35,550 questions across multiple domains. RankLLM achieves 90% agreement with human judgments and consistently outperforms strong baselines such as IRT. It also exhibits strong stability, fast convergence, and high computational efficiency, making it a practical solution for large-scale, difficulty-aware LLM evaluation.

1 INTRODUCTION

Large Language Models (LLMs) have shown impressive breadth across tasks from natural language understanding to multi-step problem solving (Young et al., 2018; Thirunavukarasu et al., 2023; Hadi et al., 2023; Kaddour et al., 2023; Zhou et al., 2023). As capability grows, reliable evaluation becomes the community's dashboard. However, popular benchmarks—e.g., MMLU-Pro (Wang et al., 2024) and MATH (Hendrycks et al., 2021b)—typically collapse performance to *accuracy within topical categories*, implicitly treating all items as equally informative. This masks differences that hinge on *difficulty* and can flip model rankings when the mixture of easy vs. hard items shifts. Counting a single arithmetic fact and a multi-step calculus derivation as equally "correct" illustrates the problem: the evaluation fails to distinguish routine pattern matching from advanced reasoning.

We posit that **difficulty must be modeled explicitly at the item level**. We introduce **RankLLM**, a simple, robust, and scalable framework that jointly estimates *question difficulty* and *model competency* from observed successes/failures. RankLLM constructs a directed bipartite interaction graph between models and questions and performs **damped bidirectional score propagation**: solving a hard question carries more evidential weight for competency; failing an easy question contributes more weight to difficulty. The process converges to a unique stationary solution, yielding difficulty and competency scores that support difficulty-aware comparison at scale. Compared to Item Response Theory (IRT), RankLLM is *non-parametric*, avoids per-item logistic fits, and runs in *linear time* in the number of model–question interactions, making it practical when per-model sample sizes are small and datasets are large.

What RankLLM brings (methodological advantages). We highlight five core advantages that make RankLLM practical and reliable for today's evaluation regimes: 1) *Human alignment*: difficulty estimates reach 90% consensus with human judgments, outperforming multiple IRT baselines across datasets; 2) *Efficiency and scalability*: propagation converges in 0.006 s on consumer hardware and scales linearly with the number of model–question interactions, supporting tens of thousands of items and dozens of models; 3) *Robustness*: question-side difficulty rankings remain highly stable under removal of individual models ($\rho = 0.998$), while model-side competency rankings remain stable under dataset perturbations ($\rho > 0.99$); 4) *Sensitivity beyond accuracy*: in controlled simulations,

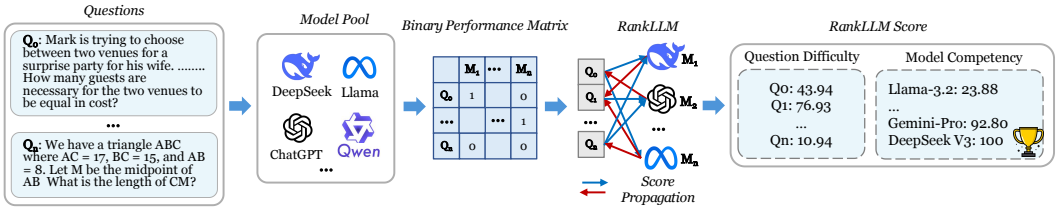


Figure 1: Schematic of RankLLM's Weighted Ranking Pipeline.

RankLLM recovers subtle performance gaps missed by accuracy- and IRT-based baselines, rewarding performance on harder items and enabling meaningful re-ordering among close models (adjacent changes consistent with Kendall's $\tau = 0.8492$); 5) *Simplicity and reproducibility*: a non-parametric graph procedure with a single damping hyperparameter; no per-item calibration or curve fitting is required, enabling easy, consistent deployments.

What RankLLM reveals (empirical findings). Applying RankLLM at scale surfaces actionable empirical patterns: 1) *Dataset-specific difficulty profiles*: MATH and MMLU-Pro exhibit broader difficulty tails suited for advanced reasoning, whereas GSM8K and HellaSwag skew easier; 2) *Model-family consistency*: families preserve characteristic difficulty patterns across parameter scales—scaling primarily shifts absolute accuracy rather than the relative difficulty structure; 3) *Open-weight model potential*: difficulty estimated on open-weight pools closely tracks full-pool results (Spearman 0.96, Pearson 0.94, Kendall 0.85), rivaling proprietary pools; 4) *Diversity benefits*: mixed-scale model pools reduce extreme misestimation by 83% relative to homogeneous pools and best align with human judgments (90% consensus); 5) *Size–performance correlation*: larger and proprietary models dominate the top competency scores, consistent with observed scaling trends.

We evaluate RankLLM across **30 models** and **35,550 questions** spanning BBH, GPQA, GSM8K, HellaSwag, MATH, and MMLU-Pro. Our study covers human alignment, convergence, robustness under model/dataset perturbations, and controlled simulations. Figure 1 illustrates the joint estimation of question difficulty and model competency.

Contributions. This work makes three key contributions: 1) **A new evaluation framework.** We introduce *RankLLM*, a difficulty-aware and non-parametric approach that jointly estimates question difficulty and model competency. This allows evaluation to move beyond flat accuracy and provide finer-grained, difficulty-sensitive comparisons of LLM performance. 2) **Comprehensive empirical validation.** Through experiments on six benchmarks covering 35,550 questions and 30 models, RankLLM demonstrates strong alignment with human difficulty judgments (90% agreement), scalability to large evaluation settings, and robustness to changes in the model pool and dataset composition. 3) **New insights into models and datasets.** By applying RankLLM at scale, we reveal distinctive dataset difficulty profiles, consistent family-level patterns across model scales, and the reliability of open-weight and mixed-scale pools for producing stable difficulty estimates, offering practical guidance for benchmark design and model selection.

2 RankLLM

Method overview. Question difficulty plays a central role in evaluating model performance. However, difficulty is inherently abstract and cannot be precisely defined. To address this, we operationalize difficulty through model failure: a question is considered difficult if even competent models are unable to solve it. RankLLM then evaluates question difficulty and measures model competency through an interactive process. RankLLM formulates questions and models as interdependent nodes within a directed bipartite graph, where questions' difficulty scores propagate to models that successfully solve questions, and models' competency score contributes to questions that they fail to answer correctly. We model this interconnection as an ergodic Markov chain defined on the bipartite graph.

We prove that this iterative process converges asymptotically to a unique stationary distribution, yielding a set of scores: π_m for model competency and π_q for question difficulty. A higher π_m value signifies a model that consistently solves challenging questions, while a higher π_q value indicates a question remains difficult even for competent models. These scores are mutually dependent, providing a self-reinforcing assessment of both models and questions.

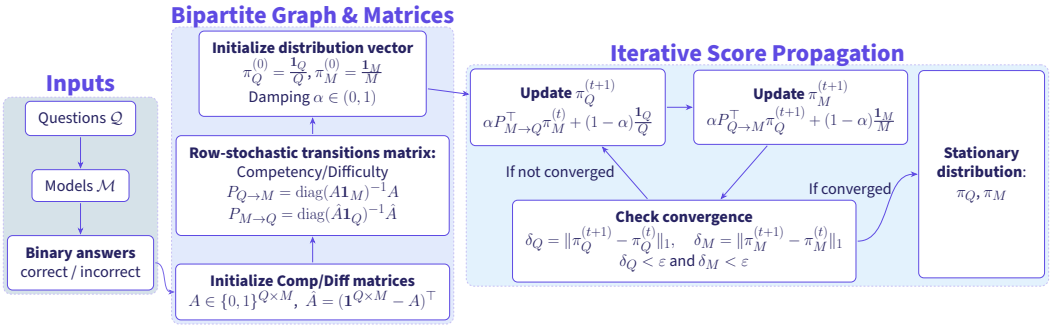


Figure 2: Detailed process demonstration of score propagation in RankLLM, which includes Evaluation Inputs, Bipartite Graph & Matrices, and Iterative Score Propagation.

2.1 Score in RankLLM

We define the difficulty score π_q for questions and the competency score π_m for models based on the stationary distribution of a coupled random walk process. This formulation reflects the intuition that question difficulty is not an inherent property but rather emerges from the opposing patterns of model successes and failures. To enable bidirectional propagation between models and questions, we establish two reciprocal information flows. First, when model m successfully answers question q, the difficulty weight of q is distributed to m in proportion to q's total solvers. Conversely, when model m fails to answer question q, its competency weight is transferred to q in proportion to the total number of failures m has encountered. Thus, the model competency π_m and question difficulty π_q are defined as:

$$\pi_m \propto \sum_{q\in \text{Success}(m)} \frac{\pi_q}{S(q)} \qquad \pi_q \propto \sum_{m\in \text{Fail}(q)} \frac{\pi_m}{F(m)} \tag{1}$$

where $S(q)$ represents the number of models that correctly solved q, and $F(m)$ denotes the total number of questions that model m has failed.

2.2 Directed Bipartite Graph and Performance Matrices

We formalize LLM-question interactions using a directed bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertex set $\mathcal{V} = \mathcal{M} \cup \mathcal{Q}$ comprises M models $\mathcal{M} = \{m_1, \ldots, m_M\}$ and Q questions $\mathcal{Q} = \{q_1, \ldots, q_Q\}$ (typically $M \ll Q$). The edge set $\mathcal{E} = \mathcal{E}_{\text{Comp}} \cup \mathcal{E}_{\text{Fail}}$ signifies model performance: **Competency Edges ($\mathcal{E}_{\text{Comp}}$):** A directed edge $(q_i \to m_j) \in \mathcal{E}_{\text{Comp}} \subseteq \mathcal{Q} \times \mathcal{M}$ indicates that model m_j correctly answered question q_i. **Difficulty Edges ($\mathcal{E}_{\text{Fail}}$):** Conversely, a directed edge $(m_j \to q_i) \in \mathcal{E}_{\text{Fail}} \subseteq \mathcal{M} \times \mathcal{Q}$ indicates that model m_j failed to answer question q_i.

These interactions are encoded into two complementary binary matrices, constructed after filtering questions: **Competency Matrix (A):** This matrix $A \in \{0,1\}^{Q\times M}$ has entries $A_{ij} = \mathbb{I}[(q_i \to m_j) \in \mathcal{E}_{\text{Comp}}]$, where $A_{ij} = 1$ if model m_j correctly answered question q_i. **Difficulty Matrix ($\hat{A}$):** This matrix $\hat{A} \in \{0,1\}^{M\times Q}$, expressed as $\hat{A} = (\mathbf{1}^{Q\times M} - A)^\top$, is the transposed complement of A. Its entries $\hat{A}_{ji} = \mathbb{I}[(m_j \to q_i) \in \mathcal{E}_{\text{Fail}}]$ indicate if model m_j failed question q_i.

Prior to matrix construction, we exclude questions universally solved by all models or failed by all models, ensuring $0 < \sum_{j=1}^{M} A_{ij} < M$ for all retained questions. This step guarantees graph connectivity and avoids trivial solutions. Such extreme cases are rare (e.g., in our experiments with 35,550 questions and 30 models, only 2% of questions were universally solved or failed, while no models showed 100%/0% accuracy) and are assigned conceptual lowest/highest difficulty. These matrices map the graph's edges to linear operators.

2.3 Propagating Scores via Iterative Refinement

Having established the directed bipartite graph, we formalize the score propagation. RankLLM employs an iterative refinement process to mutually determine question difficulty scores (π_Q) and model competency scores (π_M). The core of this process relies on two row-stochastic transition matrices:

Competency transition ($P_{Q\to M}$): $P_{Q\to M} = \mathrm{diag}(A\mathbf{1}_M)^{-1}A$. Given a question q, the probability of transitioning to a model m is proportional to m's success on q. The q-th row is normalized by $S(q)$, the number of models that correctly solved q.

Difficulty transition ($P_{M\to Q}$): $P_{M\to Q} = \text{diag}(\hat{A}\mathbf{1}_Q)^{-1}\hat{A}$. Given a model m, the probability of transitioning to a question q is proportional to m's failure on q. The m-th row is normalized by $F(m)$, the number of questions m failed.

Scores are updated iteratively. An underlying random walk on the bipartite graph ($\mathcal{Q} \leftrightarrow \mathcal{M}$) would be 2-periodic. To ensure convergence to a unique stationary distribution, we introduce a damping factor $\alpha \in (0, 1)$, representing a "teleportation" probability. The iterative update equations for the scores at step $t + 1$ are:

$$\pi_Q^{(t+1)} = \alpha P_{M\to Q}^\top \pi_M^{(t)} + (1-\alpha)\frac{\mathbf{1}_Q}{Q} \quad (2) \qquad \pi_M^{(t+1)} = \alpha P_{Q\to M}^\top \pi_Q^{(t+1)} + (1-\alpha)\frac{\mathbf{1}_M}{M} \quad (3)$$

Here, $\mathbf{1}_Q$ and $\mathbf{1}_M$ are all-ones vectors of appropriate dimensions, $Q = |\mathcal{Q}|$, and $M = |\mathcal{M}|$. The transposes $P_{M\to Q}^\top$ and $P_{Q\to M}^\top$ facilitate score propagation. $\pi_M^{(t+1)}$ uses the just-updated $\pi_Q^{(t+1)}$.

This iterative process, where competency and difficulty scores are mutually reinforced, is guaranteed to converge to unique stationary distributions π_Q and π_M. This is because the damped system forms an ergodic Markov chain, whose properties are established by the Perron-Frobenius theorem (Perron, 1907). A formal proof of existence, uniqueness, and convergence is provided in Appendix C.

2.4 Extending RankLLM to Continuous Scores

Benchmarks that grade free-form answers often provide partial credit rather than binary outcomes. RankLLM accommodates this setting by allowing the response matrix to take values in the unit interval. Let $A_c \in [0, 1]^{Q\times M}$ denote this continuous analogue of the correct–response matrix (c stands for "**c**ontinuous"). All subsequent quantities mirror the binary case after replacing A with A_c: The row sums $S(q) = \sum_{m=1}^{M}(A_c)_{qm}$ represent the total score achieved on question q, with questions satisfying $S(q) = 0$ removed as before. The complement matrix $\hat{A}_c = \mathbf{1}^{Q\times M} - A_c$ and corresponding column sums $F(m) = \sum_{q=1}^{Q}(\hat{A}_c)_{mq}$ maintain the same interpretation, with $F(m) > 0$ ensured in practice. The transition matrices retain the same form:

$$P_{Q\to M} = \text{diag}(S)^{-1}A_c \quad (4) \qquad P_{M\to Q} = \text{diag}(F)^{-1}\hat{A}_c \quad (5)$$

The iterative updates in Equation 2–3 and convergence guarantee of Theorem C.1 apply *unchanged*.

3 Experiment

3.1 Experiment Setup

Datasets & benchmarks & models. We select a diverse and representative set of datasets that encompasses diverse domains, including math, science, natural language understanding, and programming. We show the details of selected datasets in Table 1. We aggregate a diverse selection of 26 mainstream LLMs, covering a wide range of sizes from 0.5B to hundreds of billions parameters[1]. The models cover a diverse range and incorporate both open-source and proprietary models to ensure a comprehensive evaluation. The details of selected models are shown in Appendix H.

Table 1: Details of selected datasets.

Dataset Name	# Questions
BBH (Suzgun et al., 2022)	6511
GPQA (Rein et al., 2023)	448
GSM8K (Cobbe et al., 2021)	1320
HellaSwag (Zellers et al., 2019)	10000
MATH (Hendrycks et al., 2021b)	5000
MMLU-Pro (Wang et al., 2024)	12102

3.2 Main Results

RankLLM offers finer-grained, difficulty-aware insights beyond traditional accuracy-based ranking. A key distinction between RankLLM and traditional accuracy-based evaluations lies in how question difficulty is incorporated. Traditional accuracy metrics treat all questions equally, which can lead to underestimating models whose overall accuracy is limited but whose strength lies in solving difficult questions. In contrast, RankLLM explicitly incorporates quantified question difficulty, enabling a more nuanced evaluation of model performance.

The positive correlation between RankLLM scores and accuracy (Kendall's Tau $\tau = 0.8492$; see Appendix G) aligns with our expectations, as stronger models typically answer more questions

[1] In RankLLM, a model is categorized as small if it has fewer than 10B parameters, medium if its size ranges between 10B and 50B parameters, and large if it exceeds 50B parameters.

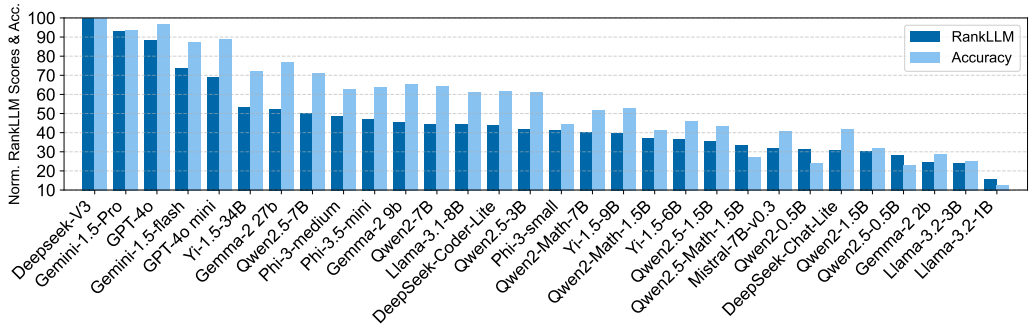


Figure 3: RankLLM scores and accuracies of models, both normalized to a fixed maximum of 100 to facilitate direct comparison. The full names of the models are provided in Appendix H.

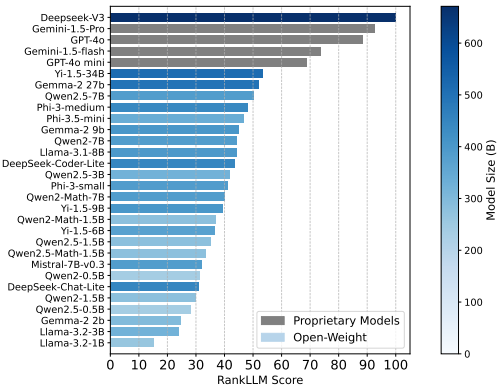


Figure 4: Relationship between model performance and parameter scale.

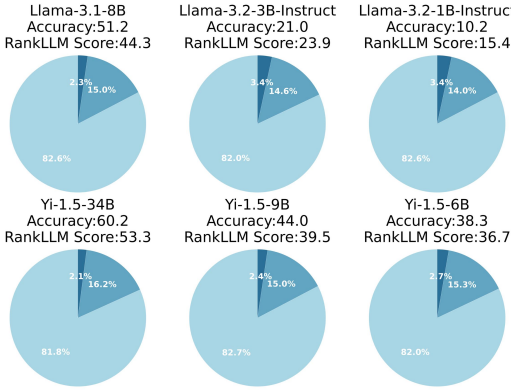


Figure 5: The proportion of Easy/Medium/Hard questions within correctly answered samples across Llama/Yi variants.

correctly and consequently receive higher scores. This validates that RankLLM serves as an enhancement to traditional accuracy-based ranking rather than a complete departure from it. However, Figure 3 reveals that significant discrepancies in model rankings frequently emerge, particularly among adjacent models. While the overall trend preserves the general hierarchy between clearly superior and inferior models, substantial rank changes frequently occur between closely-performing models, highlighting RankLLM's ability to provide finer-grained differentiation. In particular, models with high raw accuracy may receive lower RankLLM scores than their neighbors, while some lower-accuracy models are ranked higher—reflecting their stronger performance on more difficult questions. For instance, Qwen2-0.5B (20.2%) is ranked higher than DeepSeek-Chat-Lite (30.49%). This re-ranking stems from RankLLM's central design: assigning greater credit to correct answers on more challenging questions. Figure 11 supports this observation: Qwen2-0.5B correctly answered 5.5% of hard questions, compared to only 2.4% for DeepSeek-Chat-Lite, which substantially boosts its RankLLM score. A similar pattern is observed among top-performing models: while GPT-4o achieves higher overall accuracy, Gemini-1.5-Pro receives a higher RankLLM score due to answering 5% more medium- and high-difficulty questions. This illustrates how RankLLM identifies strengths on difficult questions that are otherwise obscured by flat accuracy scores—a finding consistent with our simulated Case Study (section 4).

Model families maintain consistent difficulty patterns despite scaling effects. As shown in Figure 5, our leave-one-out[2] evaluation demonstrates that models within the Llama-3 family (Llama-3.1-8B, Llama-3.2-3B-Instruct, Llama-3.2-1B-Instruct) exhibit stable difficulty-specific answering distributions across different parameter scales. Despite substantial differences in overall accuracy ($51.2\% \rightarrow 21.0\% \rightarrow 10.2\%$) and RankLLM scores ($44.3\% \rightarrow 23.9\% \rightarrow 15.4\%$), all variants preserve a similar distribution of correct responses across difficulty levels (hard / medium / easy).

[2]Leave-One-Out is a cross-validation method that iteratively uses each sample in the dataset as the validation set, while the remaining samples form the training set.

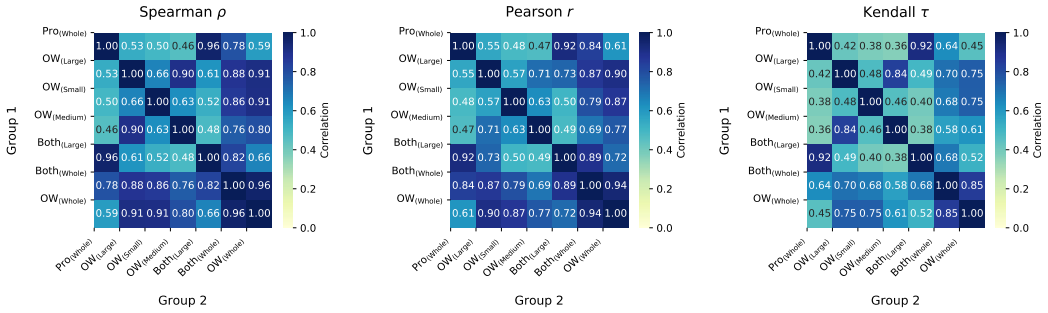


Figure 6: Correlation analysis across model group combinations (Pro. means proprietary, and OW means open weight).

Notably, the 8B model demonstrates a slight disadvantage in hard-question accuracy (11.1% compared to 15.7% and 15.6% for the smaller variants), yet maintains nearly identical easy-question accuracy (62.9% compared to 60.4% and 61.5% for the smaller variants). This trend extends across other model families, including Qwen and Yi (Appendix F), indicating that: 1) Accuracy scaling primarily influences absolute performance rather than altering relative difficulty distributions. 2) Traditional accuracy metrics obscure essential behavioral consistencies across model scales.

Open-weight models show strong potential in estimating difficulty, rivaling proprietary models. To evaluate the consistency of question difficulty distributions across model groups, we employed three correlation methods: Spearman, Pearson, and Kendall. These methods, capturing monotonic, linear, and rank-based correlations respectively, yielded highly consistent results, indicating strong agreement in the difficulty rankings produced by different model sets. As shown in Figure 6, the difficulty distributions derived from various model groups exhibit significant correlations. The difficulty distributions generated by open-weight models of all sizes ($\text{OW}_{\text{(Whole)}}$) demonstrate a strong positive correlation with those generated by the entire set of models ($\text{Both}_{\text{(Whole)}}$). Quantitatively, the Spearman, Pearson, and Kendall correlations are 0.96, 0.94, and 0.85, respectively. These high correlation values across all three metrics indicate a strong agreement between the difficulty rankings produced by open-weight models and those produced by the full model set, suggesting that open-weight models alone are capable of capturing the overall difficulty landscape as represented by RankLLM. The difficulty distributions derived from all proprietary models ($\text{Pro}_{\text{(Whole)}}$) show a correlation with those derived from all models ($\text{Both}_{\text{(Whole)}}$), with a correlation of 0.78, 0.84, and 0.64. While still demonstrating agreement, there is a noticeable drop in the Spearman correlation compared to the open-weight models.

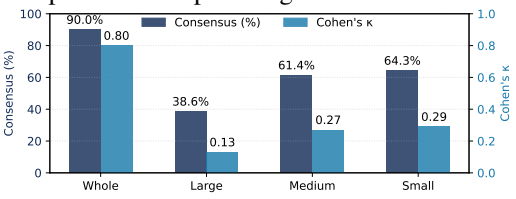


Figure 7: Consensus alignment with human on RankLLM difficulty by model size group.

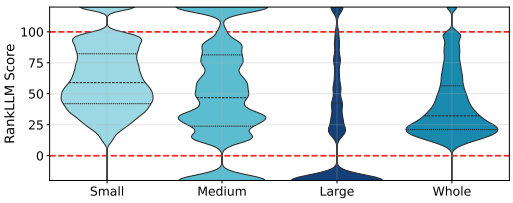


Figure 8: Difficulty distribution grouped by model size.

A diverse model selection in RankLLM mitigates extremes in difficulty estimation of questions. As shown in Figure 8[3], we analyze the impact of model diversity on difficulty estimation by evaluating RankLLM using four different annotator model pools: small-only, medium-only, large-only, and a combination of all (Whole). Our findings highlight that the composition of the models in RankLLM plays a crucial role in determining the quality of difficulty estimation. Homogeneous model groups tend to exhibit more extreme difficulty distributions. Specifically, the Large group classifies 58% of questions as "overly easy", while the Small and Medium groups contain over 30% "impossible" questions. These extremes introduce a high proportion of "dead nodes" in RankLLM's algorithm, where models either consistently overestimate or underestimate question difficulty. In contrast, the Whole group, which includes models of varying scales, achieves a more balanced difficulty distribution. Complementary error patterns across different model scales reduce extreme cases to

[3]Area above 100 represents questions that all models failed, below 0 are all correct answers.

Table 2: Alignment analysis with human evaluation on question difficulty (all metrics in %).

Method	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	$\cdots$	V_{20}	Consensus (%)
Simple Rank	41.4	54.3	50.0	60.0	51.4	54.3	54.3	50.0	50.0	$\cdots$	52.9	62.9
1PL-IRT	37.1	44.3	40.0	52.9	51.4	44.3	45.7	45.7	48.6	$\cdots$	41.4	50.0
2PL-IRT	35.7	45.7	41.4	54.3	52.9	42.9	44.3	47.1	47.1	$\cdots$	42.9	51.4
Multi-IRT	**50.0**	50.0	48.6	54.3	48.6	57.1	47.1	47.1	44.3	$\cdots$	48.6	52.9
RankLLM	37.1	**70.0**	**62.9**	**71.4**	**67.1**	**61.4**	**74.3**	**64.3**	**57.1**	$\cdots$	**62.9**	**90.0**

below 3%. Moreover, the mixed-scale ensemble maintains a valid gradient flow in RankLLM's bidirectional ranking algorithm, reducing dead nodes by 83% compared to homogeneous groups. These results align with the "wisdom of the crowd" principle, demonstrating that incorporating diverse models not only mitigates mutual biases but also preserves sufficient granularity to prevent systematic overestimation or underestimation in difficulty assessment.

Difficulty distribution varies by dataset. As shown in Figure 9, the six benchmarks exhibit distinct difficulty profiles. GPQA has a relatively uniform distribution, making it ideal for evaluating performance across a range of difficulty levels. In contrast, GSM8K, HellaSwag, and BBH are skewed toward easier questions, with most items concentrated at lower RankLLM scores. This skew may limit their utility in assessing performance on harder tasks but provides insights into models' fundamental capabilities. MATH and MMLU-Pro, on the other hand, display broader and more dispersed distributions, with MATH emphasizing harder questions and MMLU-Pro displaying a bimodal pattern, blending challenging items. These benchmarks, therefore, are better suited for evaluating advanced reasoning capabilities and the robustness of models.

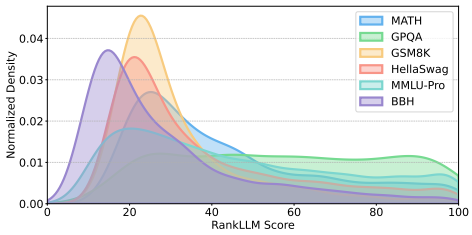


Figure 9: Question difficulty distributions for six benchmarks.

3.3 Human Evaluation

To validate the alignment of RankLLM with human judgment, we conducted a controlled blind trial involving 20 human evaluators (V_1–V_{20}). Each evaluator compared two questions from the same subject category (e.g., mathematics) to judge which was more difficult. There are 70 randomly ordered question pairs in total. A detailed evaluation protocol is in subsection H.3.

Baseline. As comparisons, we selected the Simple Rank method as a primary baseline. Additionally, we evaluated three Item Response Theory (IRT) based models: 1pl-irt, 2pl-irt, and multi-irt. Configuration details for these IRT models are provided in subsection H.4. Simple Rank determines question difficulty based solely on the number of models that answered incorrectly, assigning higher difficulty to questions with higher error counts.

RankLLM achieves high alignment (90%) with human consensus. As shown in Table 2, RankLLM achieves superior agreement with human judgments for 18 out of 20 evaluators when compared to Simple Rank, with an average margin of +9.3% in individual alignment scores over Simple Rank. The "Consensus" column in Table 2 measures agreement with the majority vote across raters for each pair, which aggregates information and resolves near-ties, showing that RankLLM achieves 90% agreement with human consensus, highlighting its strong alignment with human judgment.

More diverse model selection in RankLLM contributes to better human alignment in measuring question difficulty. As shown in Figure 7, the whole group, comprising models with various sizes, achieves a consensus alignment of 90.0% on the difficulty of the question with human evaluation, along with a Cohen coefficient of κ of 0.80. In contrast, the homogeneous size groups from the SOTA large model to the 0.5B small models exhibit markedly lower consensus (38.6%–64.3%) and weaker agreement levels ($\kappa = 0.13$–0.29). This strongly suggests that diversity in model size leads to richer and more robust difficulty assessments, reducing the bias or blind spots often observed when the annotator pool is restricted to a single-size scale.

3.4 Efficiency and Scalability Analysis of RankLLM

Given the rapidly growing number of models (M) and extensive question sets (Q) used in benchmarks, computational efficiency and scalability of the evaluation framework are paramount. We therefore analyze these aspects of RankLLM both theoretically and empirically.

RankLLM demonstrates strong computational efficiency and scalability. RankLLM has time complexity $O(tQM)$ where t is the number of iterations required for convergence (proof in Appendix D). The iteration count t remains stable with preset damping factor α and tolerance ϵ, independent of Q or M (Appendix C). Since $M \ll Q$ in typical LLM evaluation scenarios, the per-iteration cost $O(QM)$ remains manageable for large-scale applications.

Empirical validations further underscore RankLLM's efficiency and scalability. In a direct comparison on a dataset of 30 LLMs and 35,000 questions, RankLLM achieved convergence substantially faster (in 0.00597 seconds) than all Item Response Theory (IRT) baselines, including 1PL-IRT, 2PL-IRT, and Multi-IRT (Table 3). Scalability was further assessed using synthetic response matrices with Q ranging up to 1,000,000 and M up to 2,000. Across all these settings, RankLLM consistently converged in a constant number of iterations (9 in our experiments), and its average per-iteration time increased linearly with the total number of interactions $Q \times M$, aligning with its theoretical complexity (Table 4).

Table 3: Convergence speed of RankLLM on consumer-grade hardware (Intel i7).

Method	Convergence Time (s)
RankLLM	**0.00597**
1PL IRT	1782.75
2PL IRT	3787.03
Multi-IRT (3D)	18.76

These combined theoretical and empirical results confirm that RankLLM not only converges quickly but also scales efficiently with the size of the evaluation. Given the low computational overhead, deployment costs are negligible even for extreme-scale evaluations. On AWS EC2 (approximately $0.05 per vCPU-hour), RankLLM enables **cost-efficient deployment for large-scale LLM evaluation**. We have established a HuggingFace leader-board platform that supports continuous updates of new models and benchmarks, capable of handling hundreds of daily updates while maintaining cost efficiency. This makes RankLLM a practical and robust solution for ranking large numbers of language models across extensive question sets.

Table 4: RankLLM convergence time under varying numbers of models and questions.

Questions (Q)	Models (M)	$Q \times M$	Iterations (T)	Total Time (s)	Avg. Time / Iter. (s)
1,000,000	2,000	2.00×10^9	9	5.28	0.5867
1,000,000	1,000	1.00×10^9	9	3.14	0.3489
500,000	1,000	0.50×10^9	9	1.93	0.2144
1,000,000	500	0.50×10^9	9	2.18	0.2422
500,000	500	0.25×10^9	9	1.13	0.1256
500,000	250	0.125×10^9	9	0.64	0.0711
250,000	500	0.125×10^9	9	0.52	0.0578
250,000	250	0.0625×10^9	9	0.30	0.0333

3.5 Robustness and Extensibility Analysis of RankLLM

Given the rapid pace at which new models are developed and integrated into evaluation pipelines, it is crucial that evaluation systems remain both stable and extensible. To this end, we examine the robustness and extensibility of RankLLM under dynamic model pool configurations, ensuring that its scores remain consistent even as models are added. To simulate the introduction of new models, we emulate pool variation by randomly removing k models (ranging from 1 to 15) from the original set of 30 models. For each value of k, we conducted 50 independent trials and computed the mean and standard deviation of the Spearman correlation ρ between the resulting scores and those obtained with the full model pool.

Even under large-scale modifications to the model pool, RankLLM preserves stable rankings of both questions and models. This is substantiated by the results in Table 5 and visualized in Figure 10. Even when half of the model pool (15 out of 30 models) was removed—simulating a scenario where a large influx of new models suddenly enters the system—the mean ρ for question difficulty scores remained high at 0.9382 when compared to rankings from the full pool. Model competency rankings exhibited even greater resilience under the same 15-model removal condition. These findings indicate that RankLLM's relative assessments of both questions and models are largely preserved despite considerable variations in the evaluation pool's size. Concurrently, a significant practical benefit observed was the reduction in computation time, demonstrating improved efficiency with smaller model sets without a critical loss in ranking fidelity. For other detailed experiments and observations such as the removal of datasets, please refer to Appendix E.

Table 5: Impact of randomly removing k models on RankLLM's stability and computation time (averaged over 50 trials).

Models	Question Correlation		Model Correlation		Computation Time	
Removed (k)	Mean ρ	SD	Mean ρ	SD	Avg. Time (s)	% Reduction
1	0.9978	0.0021	0.9997	0.0002	0.0079	14.09
5	0.9850	0.0073	0.9988	0.0009	0.0080	13.57
10	0.9684	0.0107	0.9977	0.0019	0.0066	28.28
15	0.9382	0.0187	0.9934	0.0069	0.0062	33.02

4 Case Study: Validating Difficulty-Based Evaluation

Due to the lack of ground truth in large-scale evaluations, identifying model capabilities on difficult questions is challenging. We address this through a controlled simulation where model performance is defined by question difficulty, enabling clear assessment of how different evaluation methods capture performance differences across difficulty levels.

Simulation Setup. We created five hypothetical models (M1–M5) and 100 questions (70 easy, 21 medium, 9 hard) to assess whether evaluation methods can detect subtle performance differences among models with similar overall accuracy. Models were configured to establish ground truth ranking: M1 > M2 > M4 > M5 > M3. M1/2 had equal accuracy, but M1 answered more hard questions. M4/5 had equal accuracy, but M4 performed better on medium-difficulty items.

Baseline. We applied RankLLM alongside baseline methods to simulated responses: **1) Standard accuracy:** Raw percentage of correct answers. **2) Dataset-difficulty-weighted accuracy:** Weighted average of accuracies on "Easy" and "Hard" subsets, with weight $w_d = 1 - \bar{a}_d$ for dataset d, where $\bar{a}_d$ is average accuracy of all models on that dataset. **3) Item Response Theory ((Van der Linden, 2018)):** 1PL, 2PL, and multidimensional IRT models. Configuration details are in subsection H.4.

Table 6: Model rankings under different evaluation methods, with ✓ indicating agreement with the ground truth ranking.

Model	Solved Qs	RankLLM		Accuracy		WeightedScore		1PL-IRT		2PL-IRT		Multi-IRT		Truth
		Score	Rank	Score	Rank	Score	Rank	Score	Rank	Score	Rank	Score	Rank	Rank
M1	70/10/5	100.00	1	85	1	77.01	2	100.00	1	100.00	1	38.91	3	1
M2	70/11/4	95.50	2	85	1	78.79	1	100.00	1	84.14	2	0.00	5	2
M3	47/13/0	56.88	5	60	5	58.86	3	0.00	5	19.66	3	95.30	2	5
M4	46/15/0	59.85	3	61	3	50.28	5	3.02	3	0.00	5	100.00	1	3
M5	47/14/0	58.21	4	61	3	58.86	3	3.02	3	19.30	4	16.52	4	4
M1>M2 Correct?		✓		✗		✗		✗		✓		✗		
M4>M5>M3 Correct?		✓		✗		✗		✓		✗		✗		

Table 7: Difficulty scores of simulated questions calculated by RankLLM.

Category	Quantity	Mean	σ
Easy	70	18.24	0.45
Medium	21	73.44	1.56
Hard	9	98.59	1.34

Analysis of Simulation Results. Only RankLLM achieved rankings consistent with ground truth (Table 6). RankLLM distinguished M1 from M2 by capturing hard question performance differences and ordered M4 > M5 > M3 based on medium-difficulty distinctions. Baseline methods showed limitations: **Standard accuracy** treats all questions equally, failing to distinguish models with identical accuracy but different hard question competency. **Dataset-weighted accuracy** uses coarse-grained weighting that ignores intra-dataset variation, allowing models to score high by answering easier items within "hard" sets. **IRT-based methods** (1PL, 2PL, Multi-IRT) showed inconsistent rankings, with Multi-IRT diverging significantly from ground truth.

These findings highlight RankLLM's unique ability to capture fine-grained performance differences through instance-level difficulty estimation, while conventional methods fail due to uniform weighting assumptions, aggregation bias, or insufficient data robustness.

5 Conclusion

We introduced RankLLM, a difficulty-aware framework for evaluating large language models. By jointly modeling question difficulty and model competence, RankLLM provides more fine-grained insights than traditional accuracy-based metrics. It achieves 90% agreement with human difficulty judgments, surpasses IRT-based and heuristic baselines, and maintains stable rankings under model or dataset perturbations. Moreover, it converges efficiently in large-scale settings. These results establish RankLLM as a scalable, robust, and human-aligned alternative for LLM evaluation.

Reproducibility Statement

We are committed to ensuring that our results can be independently verified and extended. All source code and prompt templates used in our experiments are included in the supplementary materials. The repository includes: (i) a step-by-step README describing the environment setup and execution commands; (ii) hosted links to all model version and formatted benchmark datasets; and (iii) configuration files specifying random seeds, hyper-parameters, and compute resources. Human-evaluation protocols are likewise included. Together, these artifacts enable an end-to-end replication of our experiments on any machine with standard GPU resources.

Ethics Statement

RankLLM is an evaluation framework and does not generate new user-facing content. All benchmarks employed are publicly available and distributed under permissive licenses. No personally identifiable or sensitive data are processed. Nonetheless, difficulty-aware rankings could conceivably be misused to dismiss models that prioritize safety or fairness over raw competence. We therefore release our code under an open license and encourage responsible adoption that complements, rather than replaces, broader alignment assessments. We disclose all model weights evaluated and provide guidelines for reproducing the study without circumventing dataset usage terms.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yanpeng Li, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.ai, 2025. URL https://arxiv.org/abs/2403.04652.

Han Bao, Yue Huang, Yanbo Wang, Jiayi Ye, Xiangqi Wang, Xiuyin Chen, Mohamed Elhoseiny, and Xiangliang Zhang. Autobench-v: Can large vision-language models benchmark themselves? *arXiv preprint arXiv:2410.21259*, 2024.

Andrew M. Bean, Simi Hellsten, Harry Mayne, Jabez Magomere, Ethan A. Chi, Ryan Chi, Scott A. Hale, and Hannah Rose Kirk. Lingoly: A benchmark of olympiad-level linguistic reasoning puzzles in low-resource and extinct languages, 2024. URL https://arxiv.org/abs/2406.06196.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15 (3), mar 2024. ISSN 2157-6904. doi: 10.1145/3641289. URL https://doi.org/10.1145/3641289.

Dongping Chen, Yue Huang, Siyuan Wu, Jingyu Tang, Liuyi Chen, Yilin Bai, Zhigang He, Chenlong Wang, Huichi Zhou, Yiqiang Li, et al. Gui-world: A dataset for gui-oriented multimodal llm-based agents. *arXiv preprint arXiv:2406.10819*, 2024a.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080*, 2024b.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, et al. Deepseek-v3 technical report, 2024a. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI, Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, et al. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence, 2024b. URL https://arxiv.org/abs/2406.11931.

Mucong Ding, Chenghao Deng, Jocelyn Choo, Zichu Wu, Aakriti Agrawal, Avi Schwarzschild, Tianyi Zhou, Tom Goldstein, John Langford, Anima Anandkumar, and Furong Huang. Easy2hard-bench: Standardized difficulty labels for profiling llm performance and generalization, 2024. URL https://arxiv.org/abs/2409.18433.

Chujie Gao, Siyuan Wu, Yue Huang, Dongping Chen, Qihui Zhang, Zhengyan Fu, Yao Wan, Lichao Sun, and Xiangliang Zhang. Honestllm: Toward an honest and helpful large language model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Mark E Glickman. Example of the glicko-2 system. *Boston University*, 28, 2012.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36:59662–59688, 2023.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

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, 2021b. URL https://arxiv.org/abs/2103.03874.

Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, et al. Metatool benchmark for large language models: Deciding whether to use tools and which to use. *arXiv preprint arXiv:2310.03128*, 2023a.

Yue Huang, Qihui Zhang, Lichao Sun, et al. Trustgpt: A benchmark for trustworthy and responsible large language models. *arXiv preprint arXiv:2306.11507*, 2023b.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, et al. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*, 2024a.

Yue Huang, Zhengqing Yuan, Yujun Zhou, Kehan Guo, Xiangqi Wang, Haomin Zhuang, Weixiang Sun, Lichao Sun, Jindong Wang, Yanfang Ye, et al. Social science meets llms: How reliable are large language models in social simulations? *arXiv preprint arXiv:2410.23426*, 2024b.

Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, Yikai Zhang, Yuqing Yang, Ting Wu, Binjie Wang, Shichao Sun, Yang Xiao, Yiyuan Li, Fan Zhou, Steffi Chern, Yiwei Qin, Yan Ma, Jiadi Su, Yixiu Liu, Yuxiang Zheng, Shaoting Zhang, Dahua Lin, Yu Qiao, and Pengfei Liu. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai, 2024c. URL https://arxiv.org/abs/2406.12753.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, et al. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, et al. Alignbench: Benchmarking chinese alignment of large language models. *arXiv preprint arXiv:2311.18743*, 2023a.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023b.

Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *European Conference on Computer Vision*, pp. 386–403. Springer, 2025.

Alejandro Lopez-Lira and Yuehua Tang. Can chatgpt forecast stock price movements? return predictability and large language models, 2023. URL https://arxiv.org/abs/2304.07619.

Frederic M Lord and Melvin R Novick. *Statistical theories of mental test scores*. IAP, 2008.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. Sources of hallucination by large language models on inference tasks, 2023. URL https://arxiv.org/abs/2305.14552.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, and Aditya Ramesh others. Gpt-4o system card, 2024a. URL https://arxiv.org/abs/2410.21276.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, et al. Gpt-4 technical report, 2024b. URL https://arxiv.org/abs/2303.08774.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, et al. Gpt-4 technical report, 2024c. URL https://arxiv.org/abs/2303.08774.

Oskar Perron. Zur theorie der matrices. *Mathematische Annalen*, 64:248–263, 1907. URL http://eudml.org/doc/158317.

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, 2023. URL https://arxiv.org/abs/2311.12022.

Ofir Ben Shoham and Nadav Rappoport. Medconceptsqa: Open source medical concepts qa benchmark, 2024. URL https://arxiv.org/abs/2405.07348.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2023. URL https://arxiv.org/abs/2206.04615.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024a. URL https://arxiv.org/abs/2403.05530.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, et al. Gemma 2: Improving open language models at a practical size, 2024b. URL https://arxiv.org/abs/2408.00118.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8): 1930–1940, 2023.

W.J. Van der Linden. *Handbook of Item Response Theory: Three Volume Set*. Chapman & Hall/CRC Statistics in the Social and Behavioral Sciences. CRC Press, 2018. ISBN 9781351645454. URL https://books.google.com/books?id=AtNMDwAAQBAJ.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. In *NeurIPS*, 2023.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

Siyuan Wu, Yue Huang, Chujie Gao, Dongping Chen, Qihui Zhang, Yao Wan, Tianyi Zhou, Xiangliang Zhang, Jianfeng Gao, Chaowei Xiao, et al. Unigen: A unified framework for textual dataset generation using large language models. *arXiv preprint arXiv:2406.18966*, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Kai-Cheng Yang and Filippo Menczer. Accuracy and political bias of news source credibility ratings by large language models, 2024. URL https://arxiv.org/abs/2304.00228.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018. doi: 10.1109/MCI.2018.2840738.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019. URL https://arxiv.org/abs/1905.07830.

Ruohong Zhang, Yau-Shian Wang, and Yiming Yang. Generation-driven contrastive self-training for zero-shot text classification with instruction-following llm, 2024. URL https://arxiv.org/abs/2304.11872.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. Sentiment analysis in the era of large language models: A reality check, 2023. URL https://arxiv.org/abs/2305.15005.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, Pengtao Xie, Caiming Xiong, Jian Pei, Philip S. Yu, and Lichao Sun. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt, 2023. URL https://arxiv.org/abs/2302.09419.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. Dyval: Graph-informed dynamic evaluation of large language models. *arXiv preprint arXiv:2309.17167*, 2023.

Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. Dyval 2: Dynamic evaluation of large language models by meta probing agents. *arXiv preprint arXiv:2402.14865*, 2024.

A Appendix

Appendix Contents

B Related Works

Benchmarking and evaluating LLMs. Owing to the advanced capabilities of LLMs, sophisticated benchmarking is essential to comprehensively assess their performance in both general and specialized domains (Chang et al., 2024). The evaluation of LLMs spans multiple fields, beginning with traditional core natural language processing (NLP) tasks such as sentiment analysis (Lopez-Lira & Tang, 2023; Zhang et al., 2023), text classification (Yang & Menczer, 2024; Zhang et al., 2024), and natural language inference (McKenna et al., 2023). In recent years, the development of diverse benchmarks has significantly expanded aspects of evaluating LLMs, enabling more comprehensive assessments of their capabilities. For example, the MMLU (Hendrycks et al., 2021a) and MMLU-Pro (Wang et al., 2024) assess models on multi-disciplinary language understanding and reasoning, while Big-Bench Hard (BBH) (Suzgun et al., 2022) evaluates their multi-step reasoning and algorithmic reasoning abilities. The GSM8K (Cobbe et al., 2021) benchmark tests performance on grade-school-level mathematics, and HumanEval (Chen et al., 2021) measures a model's capability in algorithmic reasoning. AlignBench is designed to evaluate the alignment of Chinese LLMs (Liu et al., 2023a), and HellaSwag (Zellers et al., 2019) focuses on commonsense reasoning by requiring LLMs to select the most plausible continuation of a given context. Guo et al. (2023) and Huang et al. (2024b) assess the abilities of LLMs in science domains. With the development of agentic AI, there are many benchmarks designed to evaluate the agent-related capabilities of LLMs (Liu et al., 2023b; Chen et al., 2024b;a; Huang et al., 2023a). Some benchmarks are also focusing on evaluating the trustworthiness of LLMs (Huang et al., 2024a; Wang et al., 2023; Gao et al., 2024; Liu et al., 2025; Huang et al., 2023b). Besides capability evaluation, some recent works propose many evaluation paradigms. For instance, flexible protocols for dynamic evaluation have been advanced, exemplified by the recent initiatives DyVal (Zhu et al., 2023) and DyVal 2 (Zhu et al., 2024). Wu et al. (2024) and Bao et al. (2024) both propose a dynamic evaluation framework powered by generative models.

Difficulty measuring. Most previous works rely on human effort to classify question difficulty levels. For example, in benchmarks like MATH (Hendrycks et al., 2021b) and GPQA (Rein et al., 2023), human experts annotate questions into predefined difficulty tiers. LingOly (Bean et al., 2024), a linguistic reasoning benchmark, categorizes question difficulty into five levels by considering both semantic similarity to English and the complexity of required reasoning. The labeling process is time-consuming and lacks scalability. Unlike human labeling, OlympicArena (Huang et al., 2024c) employs GPT-4V as an annotator to categorize question difficulty levels. MedConceptsQA (Shoham & Rappoport, 2024), on the other hand, determines difficulty levels based on the relative distances among answer choices within an undirected graph built by medical code. However, these approaches either introduce bias in difficulty assessment due to reliance on a single model or are domain-specific, limiting their generalizability to other fields. Recent work, Easy2Hard-Bench (Ding et al., 2024), applied two independent methods to quantify question difficulty: Item Response Theory (IRT) (Lord & Novick, 2008) and Glicko-2 (Glickman, 2012). Unlike IRT, which assumes static difficulty parameters, RankLLM dynamically adjusts scores based on model performance, providing a more adaptive and accurate representation of difficulty. In contrast to Glicko-2, which relies on pairwise matchups, RankLLM employs a bidirectional score propagation approach, allowing question difficulty to be inferred from a broader set of solver interactions rather than isolated 'matches', which provides a dataset-wide estimate of difficulty.

C Proof of Convergence

Convergence. Let $\Delta^{(t)} = \|\pi_Q^{(t)} - \pi_Q^{(t-1)}\|_1 + \|\pi_M^{(t)} - \pi_M^{(t-1)}\|_1$ measure the change between iterations. The algorithm terminates when $\Delta^{(t)} < \epsilon$ for predefined tolerance ϵ. For irreducible chains, convergence occurs at a geometric rate $O(\alpha^t)$, typically requiring $\log \frac{1}{\epsilon}$ iterations. An important consideration is the convergence speed in large-scale settings. In our experiments with 30 models and 35550 questions, we find that about 9 iterations of the bipartite propagation process are sufficient to reach a stable distribution of difficulty and competency scores. Each iteration primarily involves matrix-vector multiplications on A and $\hat{A}$, making the approach computationally feasible even with tens of thousands of questions. The relation between the damping factor α and the speed of convergence is shown in Table 8.

Theorem C.1 (Existence and Uniqueness of the Stationary Distribution). *Let $P_{Q\to M} \in \mathbb{R}^{Q\times M}$ and $P_{M\to Q} \in \mathbb{R}^{M\times Q}$ be the row-stochastic matrices defined in subsection 2.3. For any damping*

Table 8: Δ for different damping factors (α) across iterations, indicating the difference between the stationary distribution and convergence.

Iteration	α=0.10	α=0.20	α=0.30	α=0.40	α=0.50	α=0.60	α=0.70	α=0.80	α=0.90	α=1
0	8.39e-02	1.68e-01	2.52e-01	3.37e-01	4.22e-01	5.07e-01	5.93e-01	6.80e-01	7.67e-01	8.55e-01
1	1.86e-03	7.65e-03	1.77e-02	3.24e-02	5.20e-02	7.70e-02	1.08e-01	1.44e-01	1.88e-01	2.38e-01
2	1.20e-06	2.00e-05	1.06e-04	3.48e-04	8.84e-04	1.91e-03	3.66e-03	6.49e-03	1.08e-02	1.70e-02
3	6.11e-10	4.14e-08	5.00e-07	2.97e-06	1.20e-05	3.76e-05	9.99e-05	2.34e-04	4.99e-04	9.84e-04
4	4.59e-13	1.19e-10	3.10e-09	3.17e-08	1.94e-07	8.54e-07	3.01e-06	9.02e-06	2.38e-05	5.70e-05
5	Converge	6.28e-13	3.69e-11	6.63e-10	6.26e-09	3.92e-08	1.86e-07	7.16e-07	2.36e-06	6.85e-06
6	Converge	Converge	3.41e-13	1.10e-11	1.65e-10	1.50e-09	9.78e-09	4.97e-08	2.09e-07	7.58e-07
7	Converge	Converge	Converge	1.44e-13	3.33e-12	4.42e-11	3.94e-10	2.64e-09	1.41e-08	6.37e-08
8	Converge	Converge	Converge	Converge	5.38e-14	1.03e-12	1.27e-11	1.12e-10	7.67e-10	4.30e-09
9	Converge	Converge	Converge	Converge	Converge	Converge	3.27e-13	3.79e-12	3.32e-11	2.32e-10
10	Converge	Converge	Converge	Converge	Converge	Converge	Converge	1.74e-13	1.87e-12	1.58e-11

factor $\alpha \in (0, 1)$, the iterative process defined in Equation 2 and Equation 3 converges to a unique stationary distribution (π_Q, π_M) satisfying

$$\pi_M = \alpha\, P_{Q\to M}^\top \pi_Q + (1-\alpha)\,\frac{\mathbf{1}_M}{M}, \tag{6}$$

$$\pi_Q = \alpha\, P_{M\to Q}^\top \pi_M + (1-\alpha)\,\frac{\mathbf{1}_Q}{Q}. \tag{7}$$

Furthermore, π_Q and π_M have strictly positive entries.

Proof sketch of Theorem C.1. **Step 1: Reformulation as a Single Markov Chain.** The alternating updates between π_Q and π_M in Equation 6 and Equation 7 can be unified into a single Markov chain on an augmented state space of size $Q + M$. Define the combined state vector $\pi^{(t)} = \begin{bmatrix} \pi_Q^{(t)} \\ \pi_M^{(t)} \end{bmatrix}$. The iterative updates become: $\pi^{(t+1)} = \alpha \mathcal{P}^\top \pi^{(t)} + (1-\alpha)\mathbf{v}$, where the block matrix $\mathcal{P} = \begin{bmatrix} 0 & P_{Q\to M} \\ P_{M\to Q} & 0 \end{bmatrix}$ preserves the bipartite transitions, and $\mathbf{v} = \begin{bmatrix} \frac{\mathbf{1}_Q}{Q} \\ \frac{\mathbf{1}_M}{M} \end{bmatrix}$ represents uniform teleportation.

Step 2: Damped Markov chain interpretation. The combined dynamics correspond to a Markov chain with a transition matrix:

$$T = \alpha \mathcal{P}^\top + (1-\alpha)\mathbf{v}\mathbf{1}_{Q+M}^\top,$$

where $\mathbf{1}_{Q+M}$ is the all-ones vector. At each step, the chain either follows $\mathcal{P}^\top$ (with probability α) or restarts uniformly via $\mathbf{v}$ (with probability $1-\alpha$), mirroring the damping mechanism in Equation 6–Equation 7.

Step 3: Irreducibility and aperiodicity. 1) *Irreducibility*: The teleportation term ensures every state can reach any other state in one step, as $(1-\alpha)\mathbf{v} > 0$ componentwise. 2) *Aperiodicity*: Self-transitions occur with probability at least $(1-\alpha)\min\left(\frac{1}{Q}, \frac{1}{M}\right) > 0$, breaking periodicity inherent in the bipartite structure. Thus, T is irreducible and aperiodic.

Step 4: Perron-Frobenius theorem application. By the Perron-Frobenius theorem for irreducible and aperiodic Markov chains, T has a unique stationary distribution $\pi = \begin{bmatrix} \pi_Q \\ \pi_M \end{bmatrix}$ with $\pi > 0$, satisfying $\pi = T^\top \pi$. Substituting T into this equation recovers the coupled updates in Equation 6–Equation 7, confirming (π_Q, π_M) as the unique solution. Crucially, convergence iteration t depends on the damping factor α and the desired precision ϵ, but it is **independent of the scale of the problem, i.e., the number of questions Q or models M.** □

This proof rigorously establishes Theorem C.1 using notation consistent with the main text. The use of $\mathcal{P}$, $\mathbf{v}$, and damping factor α directly corresponds to the iterative updates in Equation 6–Equation 7, ensuring notational coherence.

D Time Complexity Analysis

In this section, we calculate and prove the time complexity of RankLLM in theory.

Let Q be the number of questions and M be the number of models. The Competency Matrix is denoted by $A \in \{0,1\}^{Q\times M}$, and the Difficulty Matrix (transposed) is $\hat{A} = (\mathbf{1}^{Q\times M} - A)^\top \in \{0,1\}^{M\times Q}$. The transition matrix $P_{Q\to M} \in \mathbb{R}^{Q\times M}$ is derived from A, and $P_{M\to Q} \in \mathbb{R}^{M\times Q}$ is derived from $\hat{A}$.

Each iteration of RankLLM primarily involves two sparse matrix-vector multiplications as defined in Equation 2 and Equation 3 of the main paper. The operation $P_{M\to Q}^\top \pi_M^{(t)}$ involves multiplying the transpose of $P_{M\to Q}$ (a $Q \times M$ sparse matrix) by the M-dimensional vector $\pi_M^{(t)}$, the computational cost of this product is proportional to the number of non-zero elements in $P_{M\to Q}$ plus the dimension of the output vector, i.e., $O(\text{nnz}(P_{M\to Q}) + Q)$. Similarly, the cost of the transposed product is $O(\text{nnz}(P_{Q\to M}) + M)$.

The sum of non-zero elements across both original transition matrices, $\text{nnz}(P_{Q\to M}) + \text{nnz}(P_{M\to Q})$, represents the total number of correct and incorrect answers, which sum to QM. Specifically, $\text{nnz}(P_{Q\to M})$ is the total number of entries where $A_{ij} = 1$, and $\text{nnz}(P_{M\to Q})$ is the total number of entries where $\hat{A}_{ji} = 1$ (i.e., $A_{ij} = 0$).

Additional per-iteration operations, such as vector additions and scalar multiplications for incorporating the damping factor α and the uniform teleportation term, contribute $O(Q + M)$.

Combining these, the total complexity for one iteration is: $O(QM + Q + M) = O(QM)$

In typical scenarios the QM term dominates $Q + M$. Therefore, the per-iteration complexity is effectively $O(QM)$. Thus, each iteration runs in $O(QM)$ time. Combining the number of iterations T and the per-iteration complexity, the overall time complexity of RankLLM is:

$$O(T \cdot (QM + Q + M)) = O(\log(1/\epsilon) \cdot (QM + Q + M)) = O(tQM) \tag{8}$$

Where ϵ is the predefined tolerance value, and t is the convergence iteration count.

E Robustness Analysis

E.1 Robustness to Single/Group Model Addition

To assess RankLLM's ranking stability as the model pool composition evolves, simulating the integration of new models, we analyze its performance under perturbations. We consider scenarios analogous to incorporating a single new model or a small group of 5 new models into an existing evaluation set. This is achieved by examining the Spearman correlation (ρ) of question difficulty and model competency scores derived from reduced model pools (N-1 models via Leave-One-Out, and N-5 models via random subset removal) against the scores from the original full pool of N models. High correlations indicate that RankLLM's assessments remain consistent. The statistics for the N-5 scenario reflect averages over 10 trials. Results are summarized in Table 9.

The analysis presented in Table 9 underscores RankLLM's robust stability when its model pool composition evolves, akin to integrating new models.

Model Competency rank exhibited greater stability: The mean Spearman correlation was exceptionally high, starting at 0.9997 (SD=0.0002) with one model removed and remaining at 0.9934 (SD=0.0069) even with 15 models removed. This highlights RankLLM's capability to maintain consistent relative model rankings despite considerable variations in the composition and size of the model evaluation pool. A more intuitive visualization is shown in Figure 10.

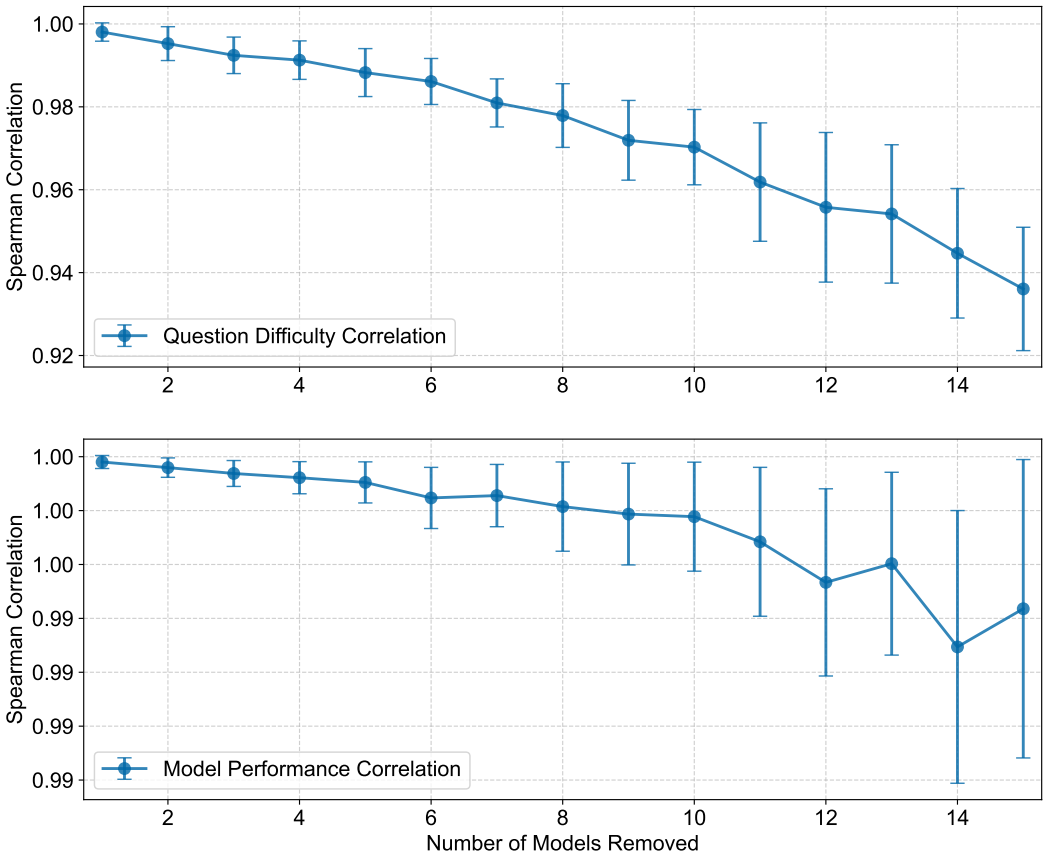


Figure 10: Impact of Random Model Subset Removal on RankLLM Stability. Top panel shows ρ for question difficulty, and the bottom panel shows ρ for model competency rank, both comparing results from the reduced model pool to the full model pool as k models are removed. Error bars represent the standard deviation over multiple random removal trials for each k.

Table 9: Comparison of RankLLM Robustness: Leave-One-Out (LOO) vs. Random Removal of 5 Models. LOO were conducted among every model in the pool(30 trials), Group removal were conducted as group with 10 trials each group. Values are averaged Spearman correlation(ρ) over respective trials.

	LOO (Remove 1)	Random Removal (Remove 5)
Question Difficulty Correlation		
Mean Spearman ρ	0.9978 ± 0.0024	0.9863 ± 0.0076
Max Observed ρ Drop	0.0100	0.0271
Model Competency Correlation		
Mean Spearman ρ	0.9998	0.9987
Question Set Reduction		
Mean Questions Removed	32.9	217.9
Max Questions Removed	103	307
Mean % Reduction	0.09%	0.62%

RankLLM maintains high consistency in question difficulty rankings despite significant model pool variations. When simulating the addition of a single model (N-1 pool correlated with the N-model pool), the Spearman correlation (ρ) for question difficulty remains exceptionally high at 0.9978 ± 0.0024. Even with a more substantial change, such as integrating a group of 5 models (N-5 pool correlated with N-model pool, a 17% pool size change), the correlation for question difficulty remains very strong at 0.9863 ± 0.0076. This indicates that the relative difficulty assessment of questions is largely preserved.

Model pool perturbations cause minimal changes to the effective question set. RankLLM filters out questions universally solved or failed by the active model pool, and Table 9 shows that adding new models (simulated via LOO or random removal) has limited effect on this filtering. Even when integrating five models, the average change was only 217.9 questions (about 0.62% of the total), with a maximum of 307. This negligible impact ensures that nearly all questions continue contributing to the evaluation, and the exclusion of this small fraction does not disrupt the overall difficulty rankings, which remain highly stable.

When viewed from the perspective of an evolving model pool, the system demonstrates strong resilience, ensuring its evaluations are dependable as new models are incorporated into the assessment framework, with model competency rankings being particularly robust.

E.2 Robustness to Dataset-Level Perturbation

To assess the influence of adding individual benchmark datasets on the final model rankings, we performed a leave-one-dataset-out analysis. The results are shown in Table 10.

Table 10: Robustness of RankLLM model competency rankings to dataset removal. Spearman ρ correlation is calculated between rankings obtained after removing a dataset and the original rankings based on all datasets.

Dataset Removed	# Questions	Model Rank Correlation (ρ)
HellaSwag	10 042	0.9604
MMLU-Pro	12 032	0.9626
MATH	5000	0.9884
GPQA	646	0.9973
GSM8K	1319	0.9973
BBH	6511	0.9973
Average Correlation		0.9839
Standard Deviation		0.0162

RankLLM model competency rankings exhibit high overall stability across all different dataset combinations. The high average Spearman correlation between model rankings derived from N-1 datasets and the full N-dataset ranking(0.9839 ± 0.0162) indicates that the relative model hierarchy established by RankLLM is largely consistent, even when any single dataset is newly introduced to or excluded from the evaluation pool. When simulating the addition of substantial datasets like MMLU-Pro ($12,032$ questions) or HellaSwag ($10,042$ questions) to the remaining pool, the model rank correlations were observed to be more perturbed compared to adding smaller datasets. However, the correlations in these scenarios ($0.9604/0,9626$) still indicate a strong preservation of the relative model ranking.

These findings suggest that RankLLM provides model evaluations that are robust to variations in the specific suite of benchmark datasets used, irrespective of whether a large, broad dataset or a smaller, more specialized one is being integrated. This consistency points to its utility in generating more generalized assessments of model capabilities, less dependent on the idiosyncrasies of individual datasets.

F Answer Distribution Across Difficulty

As shown in Figure 11 (Highlighted in square box), a similar pattern emerged aligned with the case study in section 4: although GPT-4o answered slightly more questions correctly overall, Gemini-1.5-Pro outperformed it on difficult questions, achieving a 0.6% higher accuracy rate in this category. Consequently, despite answering 2.9% fewer medium-difficulty questions and achieving a lower overall accuracy rate, Gemini-1.5-Pro scored higher than GPT-4o under RankLLM, further emphasizing the impact of performance on high-difficulty questions. This real-world example corroborates the findings of our simulation, demonstrating RankLLM's ability to differentiate models based on their performance on challenging tasks.

G Analysis on Correlation and Difference between RankLLM and Accuracy-Based Model Rankings

While RankLLM's competency scores exhibit a strong overall correlation with traditional accuracy-based rankings (Kendall's Tau $\tau = 0.876$, $p < 0.001$, as detailed in Table 11), a closer examination reveals significant and insightful differences in how individual models and groups of models are ordered. This section delves into these inter-model ranking variations, underscoring RankLLM's ability to provide a more nuanced perspective on model capabilities than accuracy alone. As

Table 11: Summary of model ranking comparison metrics between rankLLM and accuracy.

Metric	Value
Mean Absolute Rank Change	1.60
Median Rank Change	1.00
Max Rank Change	4.00
Kendall's Tau Correlation	0.876(p<0.001)
Rank-Biased Overlap (RBO)	0.896
ICC1 (Absolute Agreement)	0.974

summarized in Table 11, the mean absolute rank change between the two methods is 1.60, with a median change of 1.0 rank position. Crucially, the maximum observed rank change for a model is 4 positions, indicating that for some models, the evaluation outcome under RankLLM can be substantially different from an accuracy-only assessment. The high Intraclass Correlation Coefficient (ICC1 = 0.974) suggest strong overall agreement in the rankings, yet the rank changes highlight the specific instances where RankLLM provides a distinct evaluation.

G.1 Distribution of Model Rank Changes

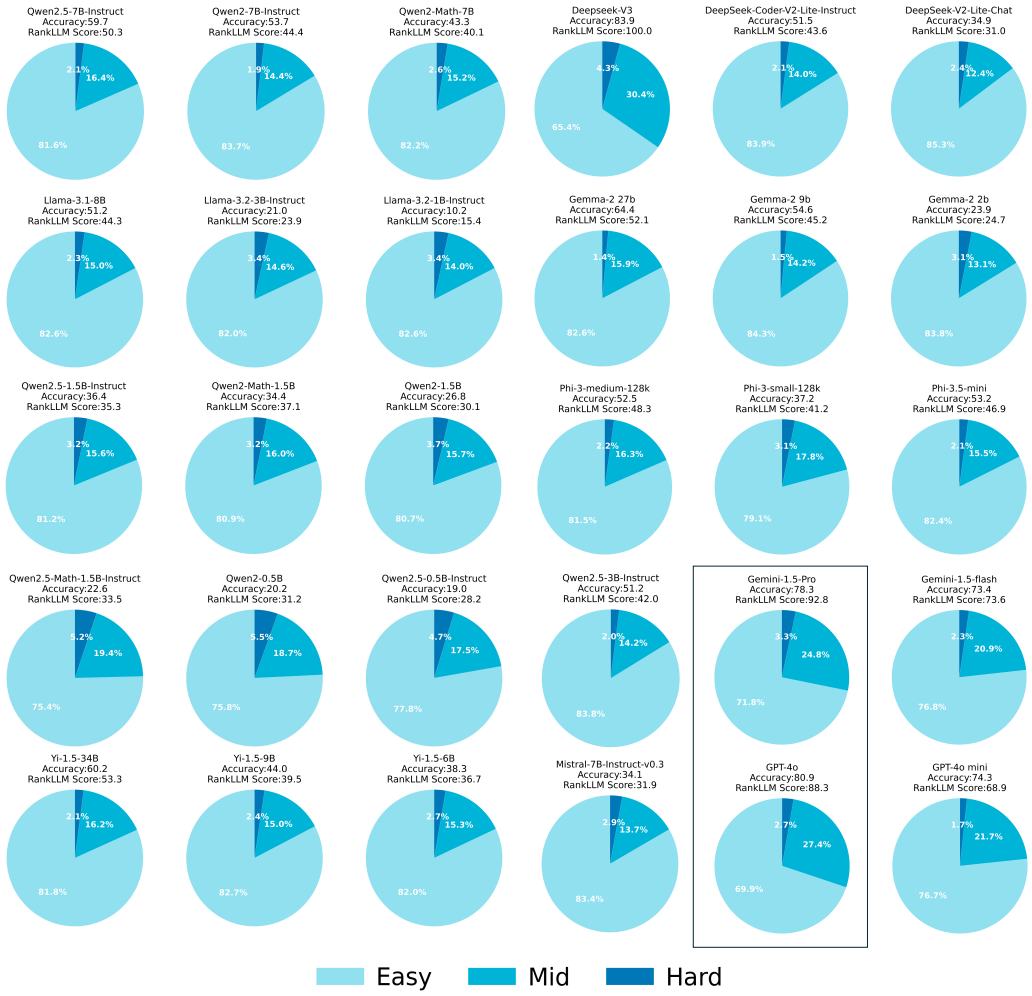


Figure 11: Distribution of correctly answered questions by models across the whole dataset (leave-one-out).

The distribution of absolute rank changes (see Figure 12) reveals that while a majority of models experience small shifts (6 models, or 20%, have no rank change, a notable portion shift by 1 or more positions. These larger shifts are particularly interesting as they point to models whose performance on questions of varying difficulty is disproportionately affecting their RankLLM score compared to their simple accuracy.

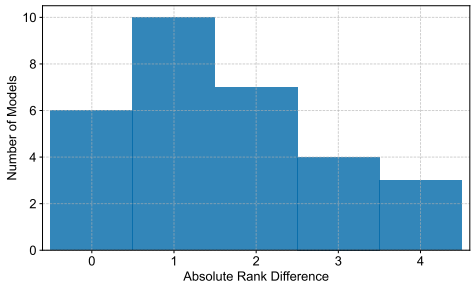
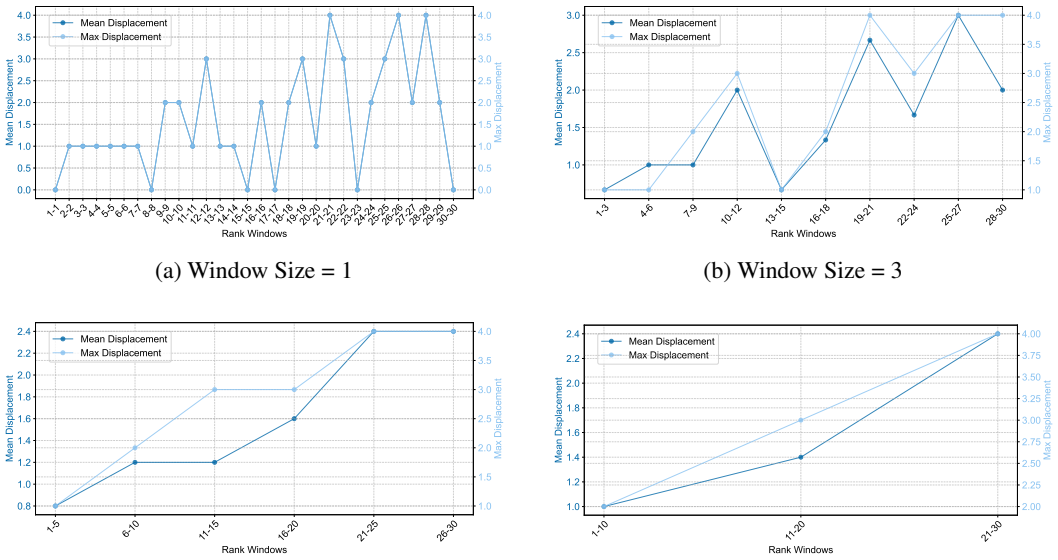


Figure 12: Absolute model rank difference distribution.

G.2 Local Rank Displacement within Accuracy-Defined Tiers

To further investigate where these ranking differences are most pronounced, we analyzed the local rank displacement of models within windows defined by their accuracy ranks. Figure 13a, Figure 13b, Figure 13c, and Figure 13d illustrate the mean and maximum rank displacement for models when grouped into tiers by their accuracy ranking, using window sizes of 1, 3, 5, and 10 respectively.



(a) Window Size = 1

(b) Window Size = 3

(c) Window Size = 5

(d) Window Size = 10

Figure 13: Mean and Maximum Rank Displacement of RankLLM scores compared to Accuracy scores within different Accuracy-Rank based windows. Windows are defined by accuracy rank (e.g., "1-10" refers to models ranked 1st to 10th by accuracy).

The trends observed in Figure 13 indicate varying degrees of rank displacement across accuracy-defined tiers:

Relatively High Stability in Top-Tier Model Rankings. For the top 10 models as ranked by accuracy (window "1-10" in Figure 13d), the mean displacement in RankLLM rank is comparatively low at 1.0, with a maximum displacement of 2.0. This suggests that among the highest-performing models by accuracy, RankLLM's rankings largely concur, although some re-ordering still occurs. The Kendall's Tau correlation within this specific window is very high ($0.867, p < 0.001$), indicating strong local rank agreement.

Increased Rank Volatility in Mid-Tier Models. The subsequent group of models, those ranked 11-20 by accuracy (window "11-20" in Figure 13d), shows increased rank displacement when evaluated by RankLLM. The mean displacement rises to 1.4, and the maximum displacement observed within this tier reaches 3.0. The local Kendall's Tau correlation ($0.733, p \approx 0.002$) remains statistically significant but is lower than that of the top tier, reflecting more substantial re-shuffling by RankLLM.

Pronounced Rank Displacement in Lower-Tier Models. Models in the lower third of the accuracy ranking (window "21-30" in Figure 13d) exhibit the highest mean displacement (2.4) and the overall maximum observed displacement of 4.0 rank positions. The local Kendall's Tau correlation drops further to 0.467 ($p \approx 0.043$), indicating considerably weaker local rank agreement between accuracy and RankLLM in this tier. This suggests that for models with lower overall accuracy, RankLLM's difficulty-weighting mechanism has a more pronounced effect, leading to more substantial re-rankings based on performance on hard questions versus reliance on easier ones.

Individual Model Rank Fluctuations Illustrate Local Reordering. When examining displacements at the individual model level (window size 1, Figure 13a), fluctuations are evident across the spectrum. For instance, the model ranked 2nd by accuracy is displaced by 1 position in RankLLM, the model ranked 9th by accuracy is displaced by 2 positions, and models ranked 21st, 26th, and 28th by accuracy are all displaced by 4 positions in their respective RankLLM rankings. This highlights that even when broader trends might align, RankLLM provides distinct local orderings based on its difficulty-aware assessment.

These windowed analyses demonstrate that while RankLLM broadly agrees with accuracy at the very top of the performance spectrum, its differentiation based on question difficulty becomes more apparent and impactful in the middle and lower tiers of accuracy rankings. This is where the ability to correctly answer challenging questions, or the failure to do so, most significantly revises a model's perceived competency compared to a simple count of correct answers. The maximum displacement of 4 rank positions underscores that RankLLM can lead to meaningfully different conclusions about relative model strengths, particularly for models with nuanced performance profiles across varying question difficulties.

H Experiment Details

To ensure optimal performance and compatibility across our experiments, we employed a combination of advanced software libraries and frameworks, including Together AI 1.2.1, OpenAI 1.30.3, vLLM 0.5.5, FlashInfer 0.1.5+cu124, Flash-Attn 2.6.3, Torch 2.4.0, Google Generative AI 0.7.2, torch 2.5.1 and CUDA 12.4.

H.1 Dataset & Prompt.

We adhered to the standard configurations outlined in the respective original benchmarks used in our experiments. For the benchmarks BBH, MMLU-Pro, GSM8k, GPQA, MATH, and HellaSwag, we utilized 5-shot prompting combined with Chain of Thought (CoT) reasoning. These strategies were chosen to align with the original settings for each respective benchmark. Temperature were set to 0 to ensure reproducibility.

H.2 Model Abbreviation.

For simplicity, we use some model abbreviations in figures and tables. Specifically, DeepSeek-Coder-Lite is DeepSeek-Coder-V2-Lite-Instruct, Qwen2.5-3B is Qwen2.5-3B-Instruct, Qwen2.5-1.5B is Qwen2.5-1.5B-Instruct, Qwen2.5-Math-1.5B is Qwen2.5-Math-1.5B-Instruct, Mistral-7B-v0.3 is Mistral-7B-Instruct-v0.3, DeepSeek-Chat-Lite is DeepSeek-V2-Lite-Chat, Qwen2.5-0.5B is Qwen2.5-0.5B-Instruct, Llama-3.2-3B is Llama-3.2-3B-Instruct, Llama-3.2-1B is Llama-3.2-1B-Instruct.

H.3 Evaluation Protocol

To ensure robust human evaluations, we implemented four key design principles: 1) Discipline selection: Varied domains (e.g., mathematics, computer science, commonsense reasoning) with matched pair disciplines. 2) Blind judgments: Evaluators were unaware of model or peer judgments.

Each of the 20 evaluators assessed 70 randomly ordered question pairs via a verification interface (Appendix J) to mitigate order bias. We computed two primary alignment metrics: 1) Individual Alignment: For each evaluator, the percentage of their non-skipped judgments (V_1–V_{20}) aligning with method predictions. 2) Consensus Alignment: The proportion of questions where a method's

Table 12: The information of selected models in RankLLM.

Model Name	Size	Developer	Version	Context Length	Open Weight
ChatGPT-4o (OpenAI et al., 2024a)	N/A	OpenAI	Aug 2024	128K	✗
ChatGPT-4o-mini (OpenAI et al., 2024c)	N/A	OpenAI	July 2024	128K	✗
DeepSeek-V3(DeepSeek-AI et al., 2024a)	671B	DeepSeek	Dec 2024	64K	✓
DeepSeek-Coder-V2-Lite-Instruct(DeepSeek-AI et al., 2024b)	15.7B	DeepSeek	June 2024	32K	✓
DeepSeek-V2-Lite-Chat(DeepSeek-AI et al., 2024b)	15.7B	DeepSeek	May 2024	32K	✓
Gemini-1.5-Pro(Team et al., 2024a)	N/A	Google	May 2024	2M	✗
Gemini-1.5-Flash(Team et al., 2024a)	N/A	Google	May 2024	1M	✗
Gemma-2-27b-it(Team et al., 2024b)	27.2B	Google	June 2024	8K	✓
Gemma-2-9b-it(Team et al., 2024b)	9.B	Google	June 2024	8K	✓
Gemma-2-2b-it(Team et al., 2024b)	2.2B	Google	June 2024	8K	✓
Llama-3.1-8B-Instruct(Grattafiori et al., 2024)	8.03B	Meta	July 2024	128K	✓
Llama-3.2-3B-Instruct(Grattafiori et al., 2024)	3.21B	Meta	Sep. 2024	128K	✓
Llama-3.2-1B-Instruct(Grattafiori et al., 2024)	1.24B	Meta	Sep. 2024	128K	✓
Mistral-7B-Instruct-v0.3(Jiang et al., 2023)	7.25B	Mistral AI	May 2024	32K	✓
Phi-3-medium-128k-instruct(Abdin et al., 2024)	14B	Microsoft	May 2024	128K	✓
Phi-3-small-128k-instruct(Abdin et al., 2024)	7.39B	Microsoft	May 2024	128K	✓
Phi-3.5-mini-128k-instruct(Abdin et al., 2024)	3.82B	Microsoft	Aug 2024	128K	✓
Qwen2-7B-Instruct (Yang et al., 2024)	7.62B	Qwen	June 2024	131K	✓
Qwen2-Math-7B-Instruct (Yang et al., 2024)	7.62B	Qwen	June 2024	131K	✓
Qwen2-1.5B-Instruct (Yang et al., 2024)	1.54B	Qwen	June 2024	131K	✓
Qwen2-Math-1.5B-Instruct (Yang et al., 2024)	1.54B	Qwen	June 2024	131K	✓
Qwen2-0.5B-Instruct (Yang et al., 2024)	0.49B	Qwen	June 2024	32K	✓
Qwen2.5-7B-Instruct (Yang et al., 2024)	7.62B	Qwen	Sep. 2024	131K	✓
Qwen2.5-Math-7B-Instruct (Yang et al., 2024)	7.62B	Qwen	Sep. 2024	131K	✓
Qwen2.5-1.5B-Instruct (Yang et al., 2024)	1.54B	Qwen	Sep. 2024	131K	✓
Qwen2.5-Math-1.5B-Instruct (Yang et al., 2024)	1.54B	Qwen	Sep. 2024	131K	✓
Qwen2.5-3B-Instruct (Yang et al., 2024)	3.09B	Qwen	Sep. 2024	131K	✓
Yi-1.5-34B-Chat (AI et al., 2025)	34.4B	01-AI	May 2024	16K	✓
Yi-1.5-9B-Chat (AI et al., 2025)	8.83B	01-AI	May 2024	16K	✓
Yi-1.5-6B-Chat (AI et al., 2025)	6.06B	01-AI	May 2024	16K	✓

prediction matches the majority of human judgment. For the set of non-skipped questions Q, this is defined as:

$$\text{Consensus} = \frac{1}{|Q|} \sum_{q \in Q} \mathbb{I}\left(\text{pred}_q = \arg\max_{j \in \mathcal{V}_q} \text{count}_q(j) \right) \tag{9}$$

Evaluator Information. Our human evaluation involved a total of 20 participants. This pool included the two authors and 18 current students not holding terminal degrees, ranging from associate-level to PhD candidates. To ensure a diverse range of perspectives, participants were recruited from various academic backgrounds, and gender balance was also considered. All evaluators possessed strong English proficiency, enabling effective task completion. Table 13 summarizes the academic fields and educational levels of the 18 student evaluators.

Table 13: Diversity of Human Evaluator Pool (N=18, excluding authors).

Academic Field	Undergraduate	Graduate (Master's)	PhD Candidate	Associate Degree	Total by Field
Computer Science	4	1	1	0	6
Other Engineering	2	1	0	0	3
Arts	1	1	0	0	2
Business	1	1	0	0	2
Humanities	3	0	0	1	4
Social Sciences	2	0	0	1	3
Total by Level	**13**	**4**	**1**	**2**	**20 (incl. 2 authors)**

Evaluators with backgrounds in computer science, mathematics, and engineering had all completed formal coursework in calculus, probability, and statistics, providing them with solid quantitative foundations. In contrast, participants from the arts, social sciences, and humanities may not have received such training. To accommodate differences in subject knowledge, participants were allowed to voluntarily skip domain-specific questions if they felt unqualified to answer them. This mechanism helped minimize the influence of unfamiliar content on the evaluation process.

Domain-specific questions requiring technical expertise were handled by participants with the appropriate background, while common sense reasoning tasks were completed by all evaluators. This design ensured that our human evaluation captured both specialized capabilities and general reasoning performance across a diverse participant pool.

H.4 IRT Baseline Configurations

To provide a robust comparison, we implemented and configured three Item Response Theory (IRT) baseline models. These configurations, detailed below, were chosen to give the IRT models ample opportunity to converge and perform optimally, even on the relatively small 100-question dataset used in our controlled simulation (section 4).

1PL IRT (Rasch Model) and 2PL IRT: Both the 1-Parameter Logistic (1PL) IRT model, also known as the Rasch model (where item discrimination is fixed at 1), and the 2-Parameter Logistic (2PL) IRT model were fitted by maximizing the log-likelihood of the observed response matrix. Optimization was performed using the L-BFGS-B algorithm, a quasi-Newton method suitable for bounded parameter estimation. A maximum of 1000 iterations was permitted for the optimization process. To enhance parameter stability and prevent extreme values, L2 regularization with a coefficient of 0.01 was applied to both model abilities (θ) and item difficulties (β). For the 2PL model, L2 regularization was additionally applied to the item discrimination parameters (α) centered around 1 (i.e., penalizing $(\alpha - 1)^2$), encouraging discriminations to be in a standard range. Parameter bounds were enforced during optimization: abilities and difficulties were constrained to the interval $[-5, 5]$, and discrimination parameters for the 2PL IRT were constrained to $[0.1, 5]$ to ensure positivity and interpretability.

Multidimensional IRT (Multi-IRT): We implemented a Multidimensional IRT (MIRT) model configured with $D = 3$ latent dimensions. This choice was made to explore a more complex latent trait structure, though we acknowledge that estimating parameters for three dimensions from a 100-question dataset can be challenging. The MIRT model was trained for 1000 epochs using the Adam optimizer with a learning rate of 0.01. Parameter estimation employed variational inference (VI), where the true posterior distributions of abilities, difficulties, and discriminations were approximated. Standard Normal distributions, $\mathcal{N}(0, 1)$, served as priors for all model parameters (abilities, difficulties, and per-dimension discrimination values). For the variational posterior approximation, we utilized a mean-field approach where the posterior for each parameter was modeled as a Normal distribution with a learnable mean and a fixed standard deviation of 1. The model was trained by maximizing the Evidence Lower Bound (ELBO). The reported ability scores for Multi-IRT are the L2 norm of the per-model multidimensional ability vectors, subsequently min-max scaled to a 0-100 range, similar to other IRT models and RankLLM scores for comparability.

I Asset Licensing and Terms of Use

This appendix details the licensing information for all software libraries, datasets, benchmarks, and language models utilized in the experiments presented in this paper. Ensuring compliance with the respective terms of use is critical for reproducible and ethical research.

I.0.1 Software Library Licenses

The software libraries employed in this research are governed by a variety of open-source licenses, predominantly permissive ones, which facilitate their use in academic and research settings. Table 14 provides a summary.

Table 14: Software Library Licenses

Library	Version	License
Together AI Python SDK	1.2.1	Apache License 2.0
OpenAI Python Library	1.30.3	Apache License 2.0
vLLM	0.5.5	Apache License 2.0
FlashInfer	0.1.5+cu124	Apache License 2.0
Flash-Attn	2.6.3	BSD 3-Clause
Torch (PyTorch)	2.4.0	BSD 3-Clause
Google Generative AI SDK	0.7.2	Apache License 2.0
torch (PyTorch)	2.5.1	BSD 3-Clause
NVIDIA CUDA Toolkit	12.4	NVIDIA CUDA Toolkit EULA

Notes on Software Library Licenses: **Together AI Python SDK 1.2.1** is licensed under Apache 2.0, as confirmed by its official repository. Community SDKs may vary. **OpenAI Python Library 1.30.3** is licensed under Apache 2.0 according to its official repository, though older PyPI versions might indicate MIT; the current repository is considered authoritative. **vLLM 0.5.5** is licensed under Apache 2.0, as per its PyPI page and official repository. **FlashInfer 0.1.5+cu124** is licensed under Apache 2.0, with the "+cu124" denoting CUDA 12.4 compatibility. **Flash-Attn 2.6.3** is licensed under BSD 3-Clause, according to its official repository. **PyTorch (Torch 2.4.0 & 2.5.1)** is licensed under BSD 3-Clause, as confirmed by its official repository and PyPI for version 2.5.1. **Google Generative AI SDK 0.7.2** is licensed under Apache 2.0, as per its official repository. Newer developments may be in 'google-genai' under the same license. **NVIDIA CUDA Toolkit 12.4** is governed by the NVIDIA CUDA Toolkit EULA, a proprietary license permitting development and specific redistribution.

I.0.2 Dataset and Benchmark Licenses

The datasets and benchmarks employed are governed by various open-source licenses or public domain dedications. Table 15 summarizes this information.

Table 15: Dataset and Benchmark Licenses

Dataset/Benchmark	Stated License(s)
BBH (Big-Bench Hard) (Suzgun et al., 2022)	Apache 2.0 (for BIG-Bench (Srivastava et al., 2023)) / MIT (for specific BBH repository by Suzgun et al.)
MMLU-Pro (Wang et al., 2024)	Apache License 2.0
GSM8k (Cobbe et al., 2021)	MIT License (likely, from original OpenAI repository)
GPQA (Rein et al., 2023)	MIT License
MATH (Hendrycks et al., 2021b)	MIT License
HellaSwag (Zellers et al., 2019)	MIT License (original author's repository)

Notes on Dataset and Benchmark Licenses: **BBH** is derived from BIG-Bench (Srivastava et al., 2023), which is under Apache 2.0. The specific repository for BIG-Bench Hard by Suzgun et al. (2022) uses an MIT license. The source utilized determines the applicable license. **MMLU-Pro** is clearly licensed under Apache 2.0 by TIGER-AI-Lab (Wang et al., 2024). **GSM8k**, as originally released by OpenAI (Cobbe et al., 2021), is likely under an MIT License. It's important to note that variants like GSM8K-Platinum may use CC-BY-4.0; the specific source license is key. **GPQA** by Rein et al. (2023) is under an MIT License. The SuperGPQA variant uses ODC-BY. **MATH** dataset by Hendrycks et al. (2021b) is clearly MIT licensed. **HellaSwag**, from the original authors' repository (Zellers et al., 2019), is under an MIT License. Other platforms hosting HellaSwag (e.g., Kaggle: CC0; Hugging Face/jon-tow: CC BY NC 4.0) may have different licenses; the original MIT license was considered authoritative for the version used.

I.0.3 Language Model Licensing and Terms of Use

Language models are subject to API service agreements or specific open-weight licenses. Table 16 provides an overview.

Notes on Language Model Licenses: **OpenAI Models'** API use is governed by OpenAI's Usage Policies and Services Agreement. Customer Content (input/output for paid API tiers) is not used for training OpenAI models, as per their Services Agreement. **Google Gemini API** is governed by Google APIs Terms of Service and the Gemini API Additional Terms of Service. Data usage for improvement depends on the service tier, according to the Gemini API Terms of Service. **Google Gemma Open Weights** are governed by the Gemma Terms of Use, allowing modification and distribution with attribution and adherence to a Prohibited Use Policy. **Meta Llama Models** are released under version-specific Llama Community License Agreements, requiring attribution and adherence to an Acceptable Use Policy (AUP). Commercial use by entities with over 700 million monthly active users typically requires a separate license, as detailed in the Llama 3 Community License, for example. **DeepSeek AI Models'** code is often MIT licensed. Some model weights are also MIT (e.g., V3-0324 release), while others are under a custom DeepSeek Model License with use restrictions. The specific license for each model must be checked. **Mistral AI Models**, such as

Table 16: Language Model Licensing and Terms Overview

Model Family/Provider	Primary Governing Terms
OpenAI (ChatGPT series, GPT-4o (OpenAI et al., 2024a), GPT-4o-mini (OpenAI et al., 2024b))	OpenAI Usage Policies & Services Agreement
Google Gemini (API: Gemini-1.5-Pro, Gemini-1.5-Flash (Team et al., 2024a))	Google APIs ToS & Gemini API Additional ToS
Google Gemma (Open Weights: Gemma-2 series (Team et al., 2024b))	Gemma Terms of Use
Meta Llama (Llama 3.1, Llama 3.2 series (Grattafiori et al., 2024))	Llama Community License Agreement & Acceptable Use Policy
DeepSeek AI (DeepSeek-V3 (DeepSeek-AI et al., 2024a), Coder-V2-Lite, V2-Lite-Chat (DeepSeek-AI et al., 2024b))	DeepSeek Model License / MIT License (varies by model)
Mistral AI (Mistral-7B-Instruct-v0.3 (Jiang et al., 2023))	Apache License 2.0
Alibaba Qwen (Qwen2, Qwen2.5 series (Yang et al., 2024))	Apache License 2.0 (most) / Qwen RESEARCH LICENSE (Qwen2.5-3B-Instruct)
01.AI Yi (Yi-1.5 series (AI et al., 2025))	Apache License 2.0
Microsoft Phi (Phi-3, Phi-3.5 series (Abdin et al., 2024))	MIT License

Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), are available under Apache 2.0. Premier models may have different licenses. **Alibaba Qwen Models** are mostly Apache 2.0. However, specific versions like Qwen2.5-3B-Instruct use a non-commercial Qwen RESEARCH LICENSE AGREEMENT. The license for each specific model should be verified. **01.AI Yi Models'** open-source releases (AI et al., 2025) are under Apache 2.0. **Microsoft Phi Models**, such as those described by Abdin et al. (2024), are generally released under the permissive MIT License.

Compliance Statement To the best of the authors' knowledge, and based on the diligent research of the licenses and terms detailed herein, the use of all software, datasets, benchmarks, and language models in this study complies with their respective governing terms and conditions. This proactive approach to documenting and adhering to licensing requirements is fundamental to responsible and ethical AI research.

Prompt I.1: Example Prompt of BBH, few-shot CoT Prompts are formed by original research team

You are a logic expert, you are given questions that involve enumerating objects and asking the model to count them.

Example 1:
<Example Question 1>
Answer:
<Example Answer 1>

Example 2:
<Example Question 2>
Answer:
<Example Answer 2>

Example 3:
<Example Question 3>
Answer:
<Example Answer 3>

Example 4:
<Example Question 4>
Answer:
<Example Answer 4>

Example 5:
<Example Question 5>
Answer:
<Example Answer 5>

1. **Answer Formatting**:
- **Multiple Choice Questions with Options**: Only select from the provided options (e.g., A, B, C, or D). If you calculate a numerical answer (e.g., "10") that matches an option, respond with the corresponding option letter (e.g., "$\boxed{A}$"), **not** the number itself. Failing to select the option will be marked as incorrect.
- **Short Answer Questions**: Provide the final answer in the format "$\boxed{X}$", where X is the correct answer. Do not include any additional formatting or explanations inside "$\boxed{}$". Do not answer only the serial number as well, Remember if answer is 1.Henry, respond with "$\boxed{Henry}$" but not "$\boxed{1}$".

2. **Answer Example**:
- For multiple choice: If the calculated answer is 10 and "10" corresponds to option C, respond with "$\boxed{C}$".
- For math questions: If the correct answer is "42," respond with "$\boxed{42}$".
- For short answer question other than math, Remember if answer is 1.Henry, respond with "$\boxed{Henry}$" but not "$\boxed{1}$".

3. **Additional Instructions**:
- Place any reasoning or calculations outside of the "$\boxed{}$" notation if necessary.
- Use "$\boxed{}$" only for the final answer.

4. **Think step by step and answer carefully**.
- You must think before answering the question.
- You must answer step by step.
Question: <question>

Prompt I.2: Example Prompt of GSM8K, CoT Prompts are sampled from original test set

You are a math expert, and you are tasked with answering questions in math with example to reference.

Example 1:
<Example Question 1>
Answer:
<Example Answer 1>

Example 2:
<Example Question 2>
Answer:
<Example Answer 2>

Example 3:
<Example Question 3>
Answer:
<Example Answer 3>

Example 4:
<Example Question 4>
Answer:
<Example Answer 4>

Example 5:
<Example Question 5>
Answer:
<Example Answer 5>

You've finished reading all the examples
Now read the question carefully and answer according to the following guidelines:

1. **Answer Formatting**:
- **Multiple Choice Questions with Options**: Only select from the provided options (e.g., A, B, C, or D). If you calculate a numerical answer (e.g., "10") that matches an option, respond with the corresponding option letter (e.g., "$\boxed{A}$"), **not** the number itself. Failing to select the option will be marked as incorrect.
- **Short Answer Questions**: Provide the final answer in the format "$\boxed{X}$", where X is the correct answer. Do not include any additional formatting or explanations inside "$\boxed{}$". Do not answer only the serial number as well, Remember if answer is 1.Henry, respond with "$\boxed{Henry}$" but not "$\boxed{1}$".

2. **Answer Example**:
- For multiple choice: If the calculated answer is 10 and "10" corresponds to option C, respond with "$\boxed{C}$".
- For math questions: If the correct answer is "42," respond with "$\boxed{42}$".
- For short answer question other than math, Remember if answer is 1.Henry, respond with "$\boxed{Henry}$" but not "$\boxed{1}$".

3. **Additional Instructions**:
- Place any reasoning or calculations outside of the "$\boxed{}$" notation if necessary.
- Use "$\boxed{}$" only for the final answer.

4. **Think step by step and answer carefully**.
- You must think before answering the question.
- You must answer step by step.
Question: <question>

J VERIFICATION TEST EXAMPLE

Prompt J.1: Verification Test Example

Welcome to RankLLM Test Program!

Please enter your First Name for identification:

===== Now at Group 1 / 100 (Group ID: 0) =====

Progress: [——————————————————] 0.0%

1. Question:
Evaluate the expression
$(751-745)+(748-742)+(745-739)+(742-736)+\cdots+(499-493)+(496-490)$.

2. Question:
What is the value of the inflection point of $f(x)=\frac{10\ln x}{x^2}$?

A. 2.000 B. 1.587 C. 0.693 D. 1.203 E. 3.014 F. 2.718 G. 4.000
H. 3.142 I. 1.000 J. 2.301

Which question is more difficult? Enter 0 if unable to judge, otherwise enter 1 or 2:

K DISCLOSURE OF LLM USAGE

We used large language models solely for editorial assistance (grammar, phrasing, and clarity). No model was used to generate technical content, derive equations, design experiments, analyze results, or write code. All datasets, algorithms, and empirical results originate from the authors' implementations and public benchmarks. No proprietary or sensitive data were submitted to third-party services.