How Many Dimensions of Intelligence? A Data Driven Clustering of LLMaaS Benchmarks

Evangelia Filiopoulou
Department of Informatics & Telematics
Harokopio University
evangelf@hua.gr

Georgia Dede
Department of Informatics & Telematics
Harokopio University
gdede@hua.gr

Anastasia Antonopoulou
Department of Informatics & Telematics
Harokopio University
it2022002@hua.gr

George Dimitrakopoulos
Department of Informatics & Telematics
Harokopio University
gdimitra@hua.gr

Mara Nikolaidou
Department of Informatics & Telematics
Harokopio University
mara@hua.gr

***Abstract*—Intelligence is the decisive criterion when selecting an LLMaaS model, since it reflects a model's ability to reason and produce accurate responses, and it stays relatively stable across providers. Yet selection is increasingly difficult, as a growing number of benchmarks each claim to measure a distinct aspect of that intelligence. This paper tests that claim. Using Spearman rank correlations across 423 models and 14 intelligence benchmarks, we find the benchmarks are highly redundant and that hierarchical clustering collapses them into three intelligence groups rather than fourteen independent ones. Grouping the benchmarks before selection removes redundancy that would otherwise distort model comparison, and we implement the procedure as a proof-of-concept decision-support tool that re-derives the groups on any dataset and ranks models within a chosen group.**

***Index Terms*—LLMaaS, intelligence, benchmarks, hierarchical clustering,interactive tool, decision making**

I. Introduction

Intelligence, understood as a model's capacity for reasoning, understanding, and accurate response generation, remains the most closely watched attribute in LLMaaS provider comparisons, since it determines whether a model can be trusted with complex tasks rather than only simple ones [1]. Organizations adopting Large Language Models as a Service (LLMaaS) face a rapidly expanding set of provider offerings, each accompanied by claims of diverse and superior intelligence. A growing number of intelligence benchmarks have been introduced in the LLM market, each claiming to capture a distinct type of capability. However, it is not clear whether many of these benchmarks actually measure independent dimensions of intelligence, or whether they simply repeat the same underlying signal under different names. This raises a direct question: how many dimensions of intelligence do these benchmarks actually measure?

This paper addresses that gap by treating benchmark redundancy as a problem to be resolved. We propose a two-phase data-driven intelligence-grouping approach that adopts hierarchical clustering, first checking benchmark redundancy and then grouping widely used LLM intelligence benchmarks into a smaller number of empirically distinct intelligence clusters. The reduced set of clusters is then made operational through an interactive tool that allows a user to upload benchmark data for any set of models, replicate the clustering procedure, and inspect model rankings within each derived intelligence category, supporting the decision-maker at the point where model selection actually takes place. The result is a decision-support tool for LLMaaS selection. It is reproducible and can be re-run as models and benchmarks change.

The rest of the paper is organized as follows: Section 2 reviews related work, Section 3 introduces LLMaaS intelligence, Section 4 describes benchmarks and their collection process. Section 5 presents the two-phase intelligence-grouping methodology, Section 6 covers the implementation, and Section 7 discusses the findings and concludes the paper.

II. Related work

Selecting an LLMaaS offering is a difficult task, and multi-criteria decision-making (MCDM) has been the usual response, since it provides a systematic way to rank alternatives against weighted criteria. In [2] LLMs are ranked with a hesitant fuzzy AHP over five trust dimensions, and in [3] a criteria set is derived for healthcare providers, covering reliability, robustness, bias and fairness, availability, performance, usability, resilience, predictability, and cost.

A second aspect of work uses clustering rather than ranking. In cloud computing, a market that resembles the LLMaaS market in structure, [4] group cloud bundles from several providers by functional attributes, namely CPU, RAM, storage capacity, and disk type, into size-based tiers that are simpler to interpret than the raw offerings. Similarly, [5] combine the Imperialist Competition Algorithm with K-means to group heterogeneous cloud workloads by cost and performance.

These two different aspects of work leave a gap. The MCDM studies treat their criteria as given, and weight them without checking whether the criteria overlap. The clustering studies group the alternatives, the services or the workloads,

979-8-3195-0225-4/26/$31.00 ©2026 IEEE

and leave the criteria as they are. Neither case matches the problem studied here, where the criteria are many and closely related. Intelligence in the LLMaaS market is reported through a growing number of benchmarks, and these benchmarks are not independent of one another. The benchmarks therefore have to be grouped first, before any selection can be made.

III. LLMaaS Intelligence

Intelligence is treated as the main criterion of this paper because it captures the model's core reasoning and understanding capacity, and its value is decisive when reasoning depth and genuine understanding matter as much as raw speed [6]. This ability is intrinsic to the model, shaped by architecture and training data, and it therefore stays relatively stable across providers and serving configurations.

Because intelligence is an abstract quality, it cannot be compared directly across models. The field has therefore operationalized it through benchmarks, each designed to capture a specific aspect of intelligence. The number of benchmarks used to evaluate LLMaaS has grown rapidly in recent years, as providers and independent research groups compete to measure ever narrower aspects of model capability, from multi-step reasoning to tool use to factual reliability.

IV. Data Collection & description

A. Data Collection

The benchmark data were obtained from Artificial Analysis [7], an independent AI benchmarking platform that provides standardized evaluations of LLMs and API providers across metrics such as intelligence, price, output speed, and latency. The dataset covers 423 models from major LLMaaS providers, spanning both proprietary models (OpenAI, Anthropic, Google, Amazon) and leading open-weight models (Meta, Alibaba, Mistral AI, ServiceNow, Cohere). All benchmark scores were collected manually in May 2026 as a single consistent snapshot, ensuring comparability across the full set.

B. Intelligence Benchmarks

We draw intelligence benchmarks from the Artificial Analysis Intelligence Index [7]. We grouped them into six categories by scope, as stated in their documentation. This is a descriptive grouping, assigned by the authors and it does not correspond to the intelligence groups derived later from the data. Below we present the corresponding benchmarks.

1) Reasoning: Reasoning benchmarks assess a model's capacity for multi-step inference over text and other complex inputs. Four benchmarks fall into this category

Humanity's Last Exam (HLE): It was developed by the Center for AI Safety and Scale AI in response to benchmark saturation. It comprises 2,500 expert-authored, closed-ended questions across more than 100 subjects, set at graduate level and requiring specialized knowledge and reasoning [8].

AA-LCR Long Context Retrieval: measures logical consistency over long inputs of roughly 100,000 tokens. The task is not retrieval of a hidden fact but synthesis and reasoning across several texts [9].

AIME 2025 Competition Math: Intelligence is measured on the 30 problems of the 2025 American Invitational Mathematics Examination. Because these problems postdate most training corpora, scores reflect mathematical reasoning rather than recall [10].

MMMU Pro: An extended version of the Massive Multi-discipline Multimodal Understanding benchmark, covering reasoning over complex images such as charts, diagrams, and chemical structures. The question is embedded in the image rather than given as text, so visual interpretation precedes any reasoning; the wider set of answer options further reduces the chance of scoring by guessing [11].

2) Scientific Knowledge: These benchmarks call for expert-level scientific understanding, drawing on domain comprehension rather than recall. Two benchmarks fall into this category.

CritPt Physics: An advanced benchmark constructed by physics researchers from unpublished problems. Solving them requires multi-step algebraic manipulation and precise numerical answers, which depend on physical understanding rather than pattern matching [12].

GPQA Diamond Scientific: The most difficult subset of the GPQA benchmark, comprising 198 multiple-choice questions in biology, physics, and chemistry. Highly educated non-experts answer the majority incorrectly even with unrestricted access to search, which makes the set a demanding test of expert-level scientific reasoning [13].

3) Coding: These benchmarks assess practical code generation on newly released problems, so performance reflects programming ability rather than recall of memorized solutions. Two benchmarks fall into this category.

LiveCodeBench Coding: A continuously updated benchmark of competitive programming problems collected as they are released, with each model scored only on problems postdating its training data [14].

SciCode: A benchmark of research-level scientific computing problems, curated by domain scientists across physics, mathematics, biology, and materials science, that evaluates applied scientific programming rather than general software development [15].

4) Agentic Tool Use: Benchmarks in this category assess agentic behavior. The model plans and executes a sequence of steps, calls external tools where needed, and is scored on whether the task is completed. The three benchmarks are described below.

Terminal-Bench Hard: A benchmark of system administration tasks carried out in a terminal environment, requiring multi-step interaction with the operating system, scripting, and error recovery [16].

Tau2 Bench Telecom Agentic Tool Use: A benchmark of agent performance on telecommunications support tasks, in which the agent calls external tools and interacts with a simulated user who can also modify the environment [17].

GDPval Agentic Real World Work Tasks: GDPval covers over a thousand occupational tasks across 44 professions, scoring models on the professional deliverables they produce,

spreadsheets, presentations, and similar work products, rather than on abstract problem solving [18]

5) Reliability: These benchmarks assess reliability of output. Whether responses are correct and whether the model withholds an answer instead of producing a confident but false one. Two benchmarks fall into this category.

AA Omniscience Accuracy Knowledge: This benchmark measures factual accuracy on questions of general encyclopedic knowledge [19].

AA Omniscience Non-Hallucination Rate: This benchmark measures how often a model declines to answer when it lacks the necessary information, rather than producing a confident but incorrect response [19].

6) Instruction Following: Benchmarks in this category measure whether a model follows instructions as specified, including constraints on format, length, and content that must be satisfied simultaneously. A single benchmark belongs to this category. *IFBench Instruction Following* evaluates a model's ability to follow complex, multi-step instructions. Unlike standard tests, it uses novel constraints. This checks whether the model actually understands the rules rather than copying familiar instruction formats [20].

V. Deriving Intelligence Groups from Benchmarks

This section sets out the procedure that reduces the 14 benchmarks to a smaller set of intelligence groups. As summarized in Fig. 1, it proceeds in two phases. The first quantifies the association among benchmarks and constructs the distance geometry on which grouping is performed. The second partitions the benchmarks by agglomerative hierarchical clustering.

Phase 1: Benchmark redundancy. When several criteria capture overlapping information, supplying them directly to a weighting procedure introduces redundancy and multicollinearity, which distorts the resulting weights [21].

Let $A = \{a_1, \ldots, a_m\}$ denote the m candidate models and $B = \{b_1, \ldots, b_p\}$ the p intelligence benchmarks, with $X^B = [x_{ij}]$ the benchmark score matrix; here $m = 423$ and $p = 14$. Because several benchmarks may measure the same underlying capability, their mutual association is examined before they are admitted as separate criteria. Association is quantified by the Spearman rank-correlation coefficient r_s, which captures monotonic relationships and is invariant to scale differences across benchmarks:

$$r_s(b_j, b_k) = 1 - \frac{c \sum_{i=1}^{m} d_i^2}{m\,(m^2 - 1)} \tag{1}$$

where d_i is the rank difference of model a_i on b_j and b_k, and $c = 6$ is the normalization constant of the Spearman rank-correlation coefficient. Applying Equation 1 to all $\binom{p}{2}$ pairs yields the correlation matrix R. The set is treated as redundant when a substantial proportion of pairs shows strong monotonic association, indicating a shared component expressed repeatedly across benchmarks.

Phase 2: Hierarchical clustering. The correlated benchmarks are partitioned into a reduced set of intelligence groups $\mathcal{K} = \{K_1, \ldots, K_k\}$, $k < p$, by agglomerative hierarchical clustering [22], [23]. The method is adopted because it requires no a priori choice of the number of clusters, which is appropriate here since the number of natural intelligence groups is unknown; k is instead selected post hoc from the dendrogram. The procedure proceeds in four steps [24].

a) Step 1: Dissimilarity matrix.: Dissimilarity is derived from Equation 1 as

$$d(b_j, b_k) = 1 - r_s(b_j, b_k) \tag{2}$$

ranging from 0 when two benchmarks rank the models identically to higher values as their orderings diverge [25]. This yields the precomputed matrix D on which all subsequent steps operate.

b) Step 2: Linkage criterion.: Clusters are merged by Average Linkage, which defines the distance between two clusters as the mean of the pairwise distances between their members.Average linkage (UPGMA) is used because it operates directly on the precomputed 1-Spearman dissimilarities and, unlike Ward's method, does not assume an underlying Euclidean space.

c) Step 3: Number of clusters and validation.: The dendrogram is cut within the largest gap between consecutive merge distances [25]. The choice is confirmed by the mean silhouette coefficient S, computed on the same matrix D.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\, b(i)\}}, \qquad S = \frac{1}{p} \sum_{i=1}^{p} s(i) \tag{3}$$

where $a(i)$ is the mean distance from i to the other members of its cluster and $b(i)$ the smallest mean distance to any other cluster. Values approaching 1 indicate cohesive, well-separated clusters; the k maximizing S and consistent with the dendrogram gap is adopted. The fidelity of the resulting hierarchy is further verified by the cophenetic correlation coefficient, the Pearson correlation between the original distances and the heights at which each pair is first joined in the tree, with values close to 1 indicating that the pairwise dissimilarities are preserved [26].

d) Step 4: Aggregation into criteria.: Cutting the validated dendrogram partitions B into disjoint groups $\mathcal{K}$ with $\bigcup_q K_q = B$. A single representative value per model and group is obtained as the arithmetic mean of normalized scores:

$$z_{iq} = \frac{1}{|K_q|} \sum_{b_j \in K_q} r_{ij}, \qquad i = 1, \ldots, m, \quad q = 1, \ldots, k \tag{4}$$

where r_{ij} is the min-max normalized score of a_i on b_j. Because all benchmarks share the common $[0, 1]$ scale, the mean is a valid aggregator. The matrix $Z = [z_{iq}]$ is the output of this phase.

VI. Results

A. Phase 1: Benchmark interdependence results

The pairwise Spearman correlations among the 14 benchmarks are shown in Figure 2. Across the 91 unique pairs

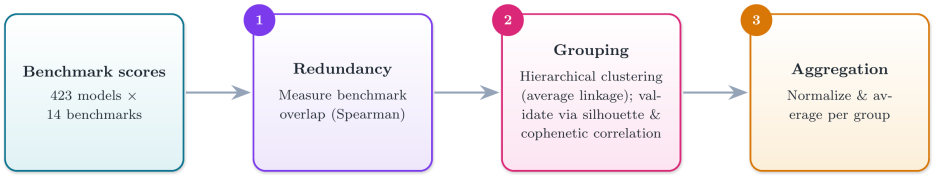


Fig. 1. Overview of the two-phase methodology. Benchmark scores for all models are first screened for redundancy via Spearman correlation (Phase 1), then grouped by agglomerative hierarchical clustering with validation (Phase 2), and finally aggregated into one representative score per intelligence group.

the median correlation is r_s= 0.55 and 62% of pairs exceed $\rho = 0.5$, so the benchmarks are clearly not independent.

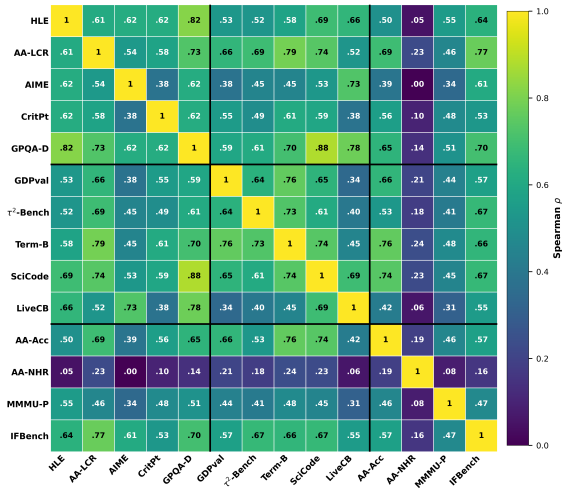


Fig. 2. Intelligence benchmarks correlations

B. Phase 2: Hierarchical Clustering Results

The resulting hierarchy is shown in the dendrogram of Figure 3. The largest gap between successive merge distances separates the within-cluster merges (0.16–0.19) from the between-cluster merges (0.48 and 0.54). Cutting within this gap, at a Spearman distance of about 0.40, yields three clusters.

The number of clusters was validated with the average silhouette coefficient. The silhouette is highest at $k = 3$ and consistent with the dendrogram gap, as shown in Figure 4.

The cluster labels were assigned after clustering, by inspecting the benchmarks that the data grouped together. Each label is therefore descriptive. The first cluster, labeled *Reasoning & Science*, groups the reasoning and science benchmarks: *Humanity's Last Exam (HLE)*, *AA-LCR Long Context Retrieval*, *AIME 2025 Competition Math*, *CritPt Physics*, and *GPQA Diamond Scientific*. These merge at very short distances, indicating that they rank the models in a similar way.

The second cluster, labeled Coding & Agentic, contains the coding and agentic benchmarks: *GDPval Agentic Real World Work Tasks*, *Tau2 Bench Telecom Agentic Tool Use*, *Terminal-Bench Hard*, *Scicode Coding*, and *LiveCodeBench Coding*. These also show tight internal merges, reflecting their shared

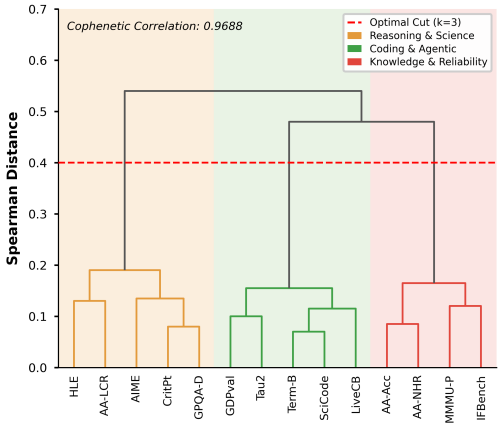


Fig. 3. Dendrogram of the 14 intelligence benchmarks

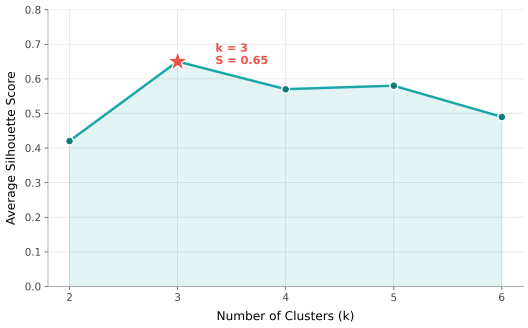


Fig. 4. Silhouette Score

reliance on code generation. The third cluster, labeled *Knowledge & Reliability*, brings together the benchmarks linked to factual knowledge and reliability: *AA Omniscience Accuracy Knowledge*, *AA Omniscience Non-Hallucination Rate*, *MMMU Pro*, and *IFBench Instruction Following*.

The cophenetic correlation coefficient of 0.9688 confirms the quality of the structure. A value this close to one indicates that the dendrogram preserves the original distances almost without distortion.

VII. Implementation

As a proof of concept, the two phases were implemented as an interactive tool, so that the framework can be applied to data beyond the snapshot used here. The procedure is fixed, but the benchmarks and models are not. New benchmarks appear



Fig. 5. Implementation

often, and the set of available models changes frequently.The tool is built in Python with a Streamlit interface, using standard scientific-computing libraries for the correlation and clustering steps.

The tool operates in three steps. First (1), the user uploads a CSV file of model names and their benchmark scores, and the tool applies the two-phase approach. Because the groups are derived from the data, their number and membership depend on the dataset supplied. Second (2), the tool presents the derived intelligence categories, so the user sees how the benchmarks group before any ranking is produced. Third (3), selecting a category displays the ranking of models within it, showing how the LLMaaS offerings are ordered inside that group. The tool reports the benchmarks in each group and leaves their labeling to the user, since the meaning of a group follows from the benchmarks it contains.

As shown in Figure 5, the user has uploaded a CSV file containing 423 models described by the aforementioned 14 intelligence benchmarks. The two-phase approach is then applied, and the intelligence dimensions are reduced to 3 clusters: Coding & Agentic, Reasoning & Science, and Knowledge & Reliability. The user then selects the first intelligence cluster, and the LLMaaS models are ranked based on their aggregated intelligence score. Specifically, Claude Opus 5 appears to lead this group, followed by Claude Mythos 5 and Fable 5. In the leaderboard, each bar is color-coded by provider, making it easy to visually distinguish models belonging to the same provider.

VIII. Conclusions-Discussion

This study shows that a wide array of benchmarks, each built to test a different aspect, collapses into a small number of distinct groups. Measured intelligence is therefore not a large set of independent dimensions but a smaller set of competencies, each captured by several correlated tests, and much of the apparent variety among benchmarks is repetition.

The 14 benchmarks, first assigned to six descriptive categories, collapsed into three. *Coding* and *agentic* benchmarks merged into one group: a model strong at code generation tends to be strong at autonomous tool use, so the two labels track one ability rather than two. Instruction following grouped with *Knowledge & Reliability* rather than standing alone, and MMMU-Pro, assigned to reasoning by scope, grouped there as well. A benchmark's stated purpose is thus not a reliable guide to the intelligence it measures. Figure 6 present an overall presentation of intelligence categorization.

These groups matter for selection. Where code and agentic performance move together, strong coding scores are a reasonable proxy for agentic competence, though the two may separate as models specialize. And where accuracy and hallucination-avoidance form their own group, high scores on hard reasoning benchmarks become a weaker guide to

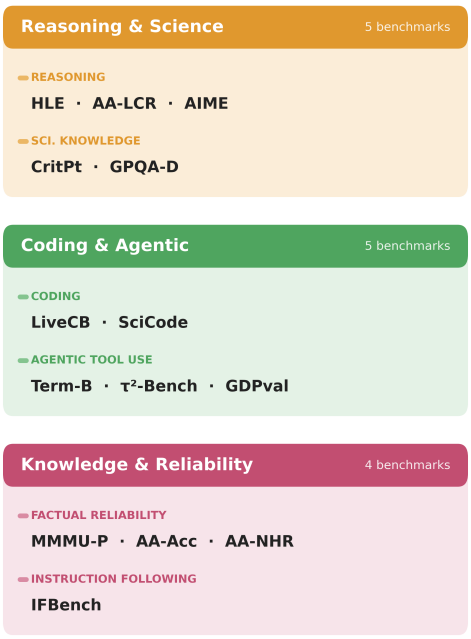


Fig. 6. Intelligence benchmarks clustering

everyday value than reliability scores. The tool builds on this. Rather than rank models on one combined score, it lets a decision-maker choose the intelligence dimension that matters for their use case and select the model that ranks highest on it.

The analysis rests on a single snapshot of 423 models and 14 benchmarks, so the groups reflect the market at one point in time. Future work should repeat it across successive snapshots to test whether the three-cluster structure is stable.

References

[1] Z. Cai, R. Ma, Y. Fu, W. Zhang, R. Ma, and H. Guan, "Llmaas: Serving large language models on trusted serverless computing platforms," *IEEE Transactions on Artificial Intelligence*, 2024.

[2] M. Aksoy, A. Adem, and M. Dagdeviren, "Trustworthiness evaluation of large language models using multi-criteria decision making," *IEEE Access*, 2025.

[3] H. M. Alabool, "Large language model evaluation criteria framework in healthcare: fuzzy mcdm approach," *SN Computer Science*, vol. 6, no. 1, p. 57, 2025.

[4] G. Fragiadakis, V. Liagkou, E. Filiopoulou, D. Fragkakis, C. Michalakelis, and M. Nikolaidou, "Cloud services cost comparison: a clustering analysis framework," *Computing*, vol. 105, no. 10, pp. 2061–2088, 2023.

[5] A. Shahidinejad, M. Ghobaei-Arani, and M. Masdari, "Resource provisioning using workload clustering in cloud computing environment: a hybrid approach," *Cluster Computing*, vol. 24, no. 1, pp. 319–342, 2021.

[6] M. N. Sakib, M. A. Islam, R. Pathak, and M. M. Arifin, "Risks, causes, and mitigations of widespread deployments of large language models (llms): A survey," in *2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*. IEEE, 2024, pp. 1–7.

[7] Artificial Analysis, "Artificial analysis: Independent analysis of AI models and API providers," https://artificialanalysis.ai, 2026, data collected manually, May 2026.

[8] L. Phan, A. Gatti, Z. Han, N. Li, J. Hu, H. Zhang, C. B. C. Zhang, M. Shaaban, J. Ling, S. Shi *et al.*, "Humanity's last exam," *arXiv preprint arXiv:2501.14249*, 2025.

[9] A. Liu, A. Mei, B. Lin, B. Xue, B. Wang, B. Xu, B. Wu, B. Zhang, C. Lin, C. Dong *et al.*, "Deepseek-v3. 2: Pushing the frontier of open large language models," *arXiv preprint arXiv:2512.02556*, 2025.

[10] M. Balunovic, J. Dekoninck, I. Petrov, N. Jovanović, and M. Vechev, "Matharena: Evaluating llms on uncontaminated math competitions," *Advances in Neural Information Processing Systems*, vol. 38, 2026.

[11] X. Yue, T. Zheng, Y. Ni, Y. Wang, K. Zhang, S. Tong, Y. Sun, B. Yu, G. Zhang, H. Sun *et al.*, "Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 15 134–15 186.

[12] M. Zhu, M. Tian, X. Yang, T. Zhou, P. Zhu, E. Chertkov, S. Liu, Y. Du, L. Yuan, Z. Ji *et al.*, "Probing the critical point (critpt) of ai reasoning: Benchmarking llms at the frontier of physics," in *APS Global Physics Summit 2026*. APS, 2026.

[13] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman, "Gpqa: A graduate-level google-proof q&a benchmark," *arXiv preprint arXiv:2311.12022*, 2023.

[14] N. Jain, A. Gu, W.-D. Li, F. Yan, T. Zhang, S. Wang, A. Solar-Lezama, K. Sen, and I. Stoica, "Livecodebench: Holistic and contamination free evaluation of large language models for code," in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 58 791–58 831.

[15] M. Tian, L. Gao, S. D. Zhang, X. Chen, C. Fan, X. Guo, R. Haas, P. Ji, K. Krongchon, Y. Li *et al.*, "Scicode: A research coding benchmark curated by scientists," *Advances in Neural Information Processing Systems*, vol. 37, pp. 30 624–30 650, 2024.

[16] M. A. Merrill, A. G. Shaw, N. Carlini, B. Li, H. Raj, I. Bercovich, L. Shi, J. Y. Shin, T. Walshe, E. K. Buchanan *et al.*, "Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces," *arXiv preprint arXiv:2601.11868*, 2026.

[17] V. Barres, H. Dong, S. Ray, X. Si, and K. Narasimhan, "Tau2-Bench: Evaluating conversational agents in a dual-control environment," *arXiv preprint arXiv:2506.07982*, 2025.

[18] T. Patwardhan, R. Dias, E. Proehl, G. Kim, M. Wang, O. Watkins, S. P. Fishman, M. Aljubeh, P. Thacker, L. Fauconnet *et al.*, "Gdpval: Evaluating ai model performance on real-world economically valuable tasks," *arXiv preprint arXiv:2510.04374*, 2025.

[19] D. Jackson, W. Keating, G. Cameron, and M. Hill-Smith, "Aa-omniscience: Evaluating cross-domain knowledge reliability in large language models," *arXiv preprint arXiv:2511.13029*, 2025.

[20] T. Zhang, Y. Hong, Y. Xia, K. Ding, Z. Zhang, Y. Wang, S. Xiang, and C. Pan, "If-bench: Benchmarking and enhancing mllms for infrared images with generative visual prompting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 8205–8215.

[21] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, and A. Song, "Efficient agglomerative hierarchical clustering," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2785–2797, 2015.

[22] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[23] F. Murtagh and P. Legendre, "Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion?" *Journal of classification*, vol. 31, no. 3, pp. 274–295, 2014.

[24] W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.

[25] E. Oti and M. Olusola, "Overview of agglomerative hierarchical clustering methods," *British Journal of Computer, Networking and Information Technology*, vol. 7, no. 2, pp. 14–23, 2024.

[26] S. Kumar and D. Toshniwal, "Analysis of hourly road accident counts using hierarchical clustering and cophenetic correlation coefficient (CPCC)," *Journal of Big Data*, vol. 3, no. 1, p. 13, 2016.