AutoJudger: An Adaptive Evaluation Framework for Efficient Benchmarking of MLLMs

Anonymous ACL submission

Abstract

Evaluating multimodal large language models (MLLMs) is becoming increasingly expensive as benchmarks grow in scale and cross-modality complexity. Inspired by structuralism in cognitive psychology, we tackle this difficulty with an adaptive evaluation framework for efficient benchmarking, namely **AutoJudger**. Instead of passively scoring on a fixed test set, AutoJudger treats evaluation as an interview-like process by keeping a hypothesized ability structure of the evaluated model and actively selecting the informative questions so as to refine these ability boundaries. Specifically, AutoJudger has three core components: ***ability decomposition*** to organize evaluation along meaningful capability dimensions, ***ability estimation*** to maintain an up-to-date quantitative profile of the model competence, and ***adaptive question selection*** to choose the most informative questions. To operationalize this paradigm, we introduce ***A*^{2}-Judger**, a novel MLLM-based **A**gentic instantiation of **A**uto**Judger** equipped with semantic-aware retrieval and dynamic memory. Experiments on four representative multimodal benchmarks show that A^2-Judger significantly improves sample efficiency while maintaining reliable evaluation results.

1 Introduction

Motivated by the success of Large Language Models (LLMs) (Achiam et al., 2023; Touvron et al., 2023; Yang et al., 2024; Liu et al., 2024a), Multimodal Large Language Models (MLLMs) (Hurst et al., 2024; Liu et al., 2023; Bai et al., 2025; Chen et al., 2025) have been developed to tackle challenging cross-modality tasks. Evaluating MLLMs typically relies on full benchmarking across diverse datasets (Li et al., 2024e; Yin et al., 2024; Li et al., 2025) that cover different aspects of capabilities (Fu et al., 2023; Liu et al., 2024d; Yue et al., 2024; Li et al., 2023, 2024d; Ge et al., 2025). However, compared to text-only settings, such evaluation under multimodal scenarios incurs extremely high costs due to long input sequences with visual contexts (Terragni et al., 2024; Xu et al., 2025), as well as computation-intensive practices such as chain-of-thought prompting (Wei et al., 2022; Guo et al., 2025; Li et al., 2024c) and LLM-assisted judging (Liu et al., 2024d; Lu et al., 2023). Consequently, such exhaustive evaluation becomes impractical under realistic constraints. This raises a fundamental question: ***Given limited evaluation budget, how can we assess MLLMs efficiently?***

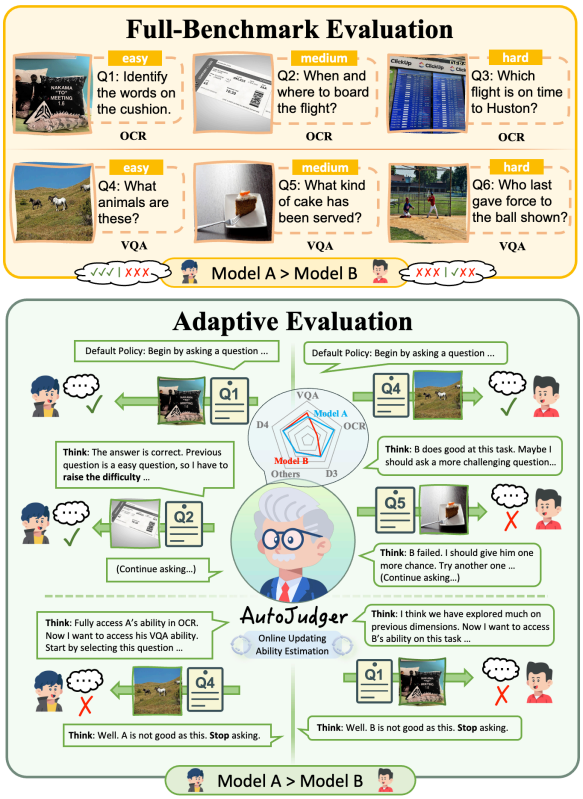


Figure 1: **Comparison between the traditional evaluation and our adaptive evaluation (AutoJudger).**

The inefficiency of traditional full-benchmark evaluation stems from its underlying assumption that evaluation samples are independent and equally informative, thereby taking the model ability as a

single aggregated quantity. However, structuralism (Piaget, 2015; Tsou, 2006) suggests that "ability" entails latent structures and is inherently multi-dimensional. Hence, evaluation samples are intrinsically connected through the underlying ability dimensions they probe. This agrees with our observations that MLLMs often exhibit highly uneven performance across different ability dimensions. As shown in Figure 1, if a model fails to solve the simplest questions from a certain dimension, repeatedly asking more questions of the same type yields little additional insight. Therefore, we believe efficient benchmarking of MLLMs should go beyond passively scoring on a fixed dataset, but instead actively select subsets by explicitly accounting for the structured nature of model ability.

Motivated by this, we propose **an adaptive evaluation framework** for efficient MLLM evaluation, namely **AutoJudger**, which maintains a set of hypothesized ability dimensions, continuously updates its beliefs about the model competence based on the evaluated model's previous performance, and adaptively selects evaluation items that are informative enough to probe the boundaries of model ability. Concretely, we decompose AutoJudger into three key modules: (1) ***ability decomposition***, which structures the ability dimensions to be assessed; (2) ***ability estimation***, which maintains an up-to-date estimate of the model ability based on the evaluation history; (3) ***adaptive question selection***, which combines the results from ability decomposition and estimation, to choose subsequent questions that explore ability boundaries over diverse decomposed dimensions.

However, instantiating AutoJudger for multimodal benchmarks poses several challenges. Firstly, ability decomposition is not always straightforward. While benchmarks such as MME (Fu et al., 2023) provide human-labeled taxonomies that explicitly define capability categories, such structured annotations are not universally available across multimodal datasets. Secondly, adaptive question selection is non-trivial. This module requires jointly balancing dimension coverage with difficulty matching, whereas prior works typically consider only one of these two factors, either through difficulty-aware sampling (Zhuang et al., 2023; Polo et al., 2024; Ding et al., 2024) or ability category-based stratification (Perlitz et al., 2023). To address these challenges, we propose **A^2-Judger**, a novel MLLM-based **A**gentic instantiation of **A**uto**Judger**. During evaluation, the judging agent acts as an interviewer, dynamically structuring ability dimensions based on explored questions, while maintaining a scorecard-form memory to track evaluation history. Combined with a semantic-aware retrieval mechanism, the agent ultimately selects questions that not only align with the ability of evaluated model but also ensure diverse coverage of dimensions. Without specific requirements for the target benchmarks, A^2-Judger can be seamlessly integrated in a plug-and-play manner.

Guided by the AutoJudger framework, we conduct a systematic review of existing baselines and compare them against A^2-Judger across 4 representative benchmarks. Our results expose the limitations of current methods and demonstrate the superiority of A^2-Judger. Notably, A^2-Judger achieves a 91% rank consistency with full-benchmark evaluation using merely 156 samples in MMT-Bench.

In summary, our contributions are threefold:

- We propose **AutoJudger**, an adaptive evaluation framework, which establishes **a novel perspective to view efficient evaluation methods** through Structuralism by structuring evaluation into ability decomposition, ability estimation, and adaptive question selection.
- To address current deficiencies identified within the AutoJudger framework, we develop an agentic instantiation, A^2-Judger. Without the need for manual ability decomposition, A^2-Judger dynamically constructs ability taxonomies during the active evaluation process.
- Experiments demonstrate that A^2-Judger generalizes effectively across benchmarks, providing the most reliable and stable assessments among all efficient evaluation methods.

2 AutoJudger

In this section, we present **AutoJudger**, our adaptive evaluation framework for efficient benchmarking of MLLMs. We first detail the general framework of AutoJudger together with its three core components in §2.1 and then introduce our MLLM-based agentic instantiation A^2-Judger in §2.2.

2.1 General Framework

Efficient benchmarking aims to reliably evaluate a model on a specific benchmark with as less expenses as possible. In this work, we constrain the

scope of evaluated models to MLLMs, due to the rapidly increasing cost of these models and the lack of prior work in this area. Let $\mathcal{Q} = \{x_i\}_{i=1}^{N}$ denote the full pool of evaluation questions (e.g., image-text pairs in a multimodal benchmark), and let $\mathcal{M} = \{M_j\}_{j=1}^{n}$ be the set of MLLMs to be evaluated. A standard static evaluation that queries all models $\mathcal{M}$ on the entire $\mathcal{Q}$, calculates scores like accuracy, and derives the ranking, can be very expensive under multimodal scenarios.

In contrast, we propose to adaptively sample a model-specific subset $\hat{\mathcal{Q}}$ for each model, leading to our evaluation framework **AutoJudger**. Inspired by the view from structuralism that abilities are latent constructs inferred from behavioral responses (Borsboom, 2005), we regard the evaluation process as an interview: by observing the model responses to selected evaluation items, we dynamically update a belief over the model's ability and then choose the next question so as to refine this belief. Although the evaluation outcome is often summarized as a single aggregate score, the underlying competence of a model is shaped by multiple latent factors (e.g., visual perception, logical reasoning, domain knowledge, and so on). Therefore, we propose to decompose the overall ability into multiple dimensions and associate each evaluation item with one or more of these dimensions. This decomposition does not claim to recover the true cognitive structure of the model, but serves as an instrumental abstraction that helps ensure coverage over diverse capability aspects. Formally, AutoJudger decomposes this "interview" process into three modular components:

Ability Decomposition This component allows AutoJudger to specify the ability dimensions D along which the model is to be assessed. These dimensions can be derived from human-defined taxonomies of important skills (e.g., task categories) or induced from the latent semantic structure of the question space (e.g., via clustering in an embedding space (Grootendorst, 2022) or applying a model to summarize). The resulting dimensions organize the benchmark into semantically or functionally coherent regions that we aim to cover.

Ability Estimation Given the ability decomposition and the history of queried questions, responses, and corresponding correctness, AutoJudger maintains an up-to-date estimate of the model ability θ. In principle, it may range from simple heuristics (e.g., smoothed accuracy) to more principled latent-trait models such as IRT (An and Yung, 2014; Cai et al., 2016), which explicitly account for both question difficulty and model performance.

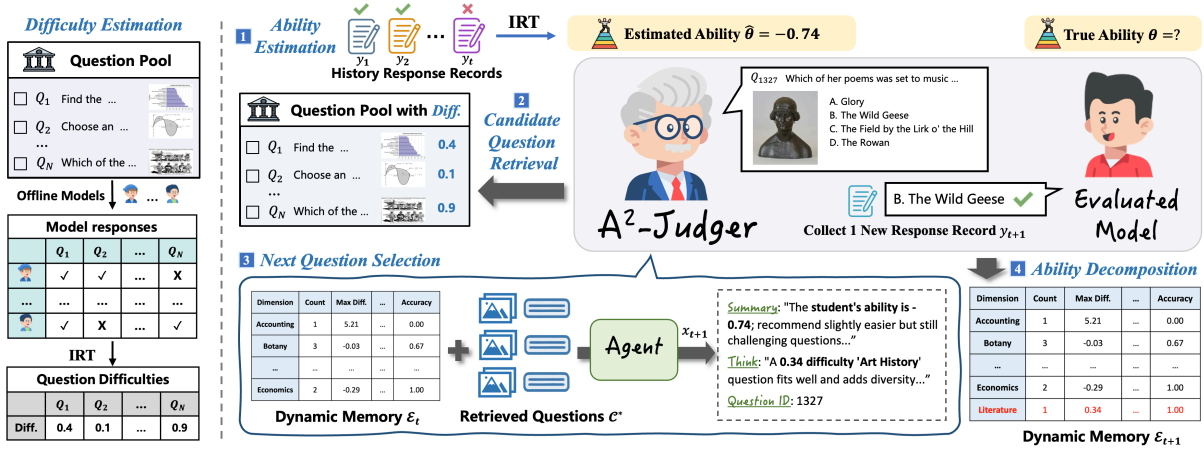
Algorithm 1 General Framework of AutoJudger

Require: Question pool $\mathcal{Q}$ with metadata (taxonomy, semantic embedding, difficulty, etc.); evaluated model M, evaluation budget B
Ensure: estimated ability θ of the evaluated model
Ensure: the selected subset of questions $\hat{\mathcal{Q}}$

1. Initialize dimensions D and ability θ
D $\leftarrow$ init_dimension($\mathcal{Q}$)
$\hat{\mathcal{Q}} \leftarrow \varnothing$
$\theta, \hat{\mathcal{Q}} \leftarrow$ init_ability_estimate(D, $\hat{\mathcal{Q}}, M$)
2. Adaptive evaluation loop
for $t = 1$ to B **do**
  $x_t \leftarrow$ adaptive_select($\mathcal{Q}/\hat{\mathcal{Q}}$, D, θ)
  $y_t \leftarrow M(x_t)$ // Query the evaluated model
  $\hat{\mathcal{Q}} \leftarrow \hat{\mathcal{Q}} \cup \{(x_t, y_t)\}$
  D $\leftarrow$ decompose_dimension(D, $\hat{\mathcal{Q}}$)
  $\theta \leftarrow$ estimate_ability(θ, D, $\hat{\mathcal{Q}}$)
  if early_stop(θ, D, t, B) **then**
    break
  end if
end for
return θ, $\hat{\mathcal{Q}}$

Adaptive Question Selection Conditioned on the current ability θ and evaluation history $(D, \hat{\mathcal{Q}})$, this module selects the next question x_t from the remaining question pool $\mathcal{Q}/\hat{\mathcal{Q}}$. As the ultimate goal is to make the most informative use of a limited evaluation budget, we explicitly consider one or more factors like dimension semantics, question difficulty, and model ability as decision signals. Concretely, the selection module can be implemented as re-defined rules, such as using the IRT model to choose questions whose difficulty best matches the estimated model ability, or always selecting the question that is semantically farthest from those already asked. Alternatively, we can also implement it as a parameterized policy, such as an MLLM-based agent that implicitly integrates all of these information as the decision basis.

The overall pipeline of our framework is summarized in Algorithm 1, where AutoJudger maintains an internal state about the estimated ability θ for the evaluated model at each step, and interacts with the benchmark through the following loop: (i) pick the next question x_t to query based on the current



(a) Before Evaluation (b) At Evaluation Iteration $t+1$

Figure 2: The agentic instantiation of AutoJudger – ***A*²-Judger**. Before evaluation, the difficulties of questions are measured by utilizing a set of offline models. At each evaluation iteration, A^2-Judger firstly retrieves the candidate questions based on the estimated ability. Then, A^2-Judger selects the most appropriate question, collects the response from the evaluated model, and updates its memory about ability dimensions and evaluation history.

state and evaluation history; (ii) observe the model response y_t and its correctness; (iii) update the ability dimensions if pre-defined taxonomies are not available; (iv) update the estimated ability.

2.2 Agentic Instantiation: A^2-Judger

By design, our AutoJudger framework has multiple instantiations. In this work, we focus on an MLLM-based agentic instantiation, where a judging agent acts as an interviewer that interacts with the evaluated model and the benchmark. The detailed evaluation process is illustrated in Figure 2. To the best of our knowledge, our work is the first to instantiate an adaptive, agent-driven framework for efficient benchmarking of MLLMs.

Difficulty-Ability Estimation Firstly, we employ the IRT model (An and Yung, 2014; Cai et al., 2016) to conduct ability estimation, which has been proved effective for modeling latent abilities (Vania et al., 2021). Since IRT-based ability estimation requires known difficulty of questions, we begin with an offline calibration stage: we collect responses from a pool of offline MLLMs on the full benchmark and fit the IRT model to obtain the difficulty for each question. During online evaluation, for the evaluated model, A^2-Judger maintains a scalar ability estimate and updates it based on the observed correctness and the calibrated question difficulty.

Ability Decomposition Considering that benchmarks do not always come with explicit human-defined taxonomies, we employ the MLLM-based judging agent to infer the semantics of questions (e.g., subject areas or skill categories) as the proxy dimensions, thereby accommodating diverse benchmark characteristics. On top of these dimensions, A^2-Judger maintains a dynamic memory $\mathcal{E}$ that aggregates evaluation history into a compact table, recording for each dimension the number of questions asked, the difficulty distribution, and empirical accuracy. This memory provides a continuously updated, interpretable, and multi-dimensional summary of the strengths and weaknesses of the evaluated model, serving as part of the context for the judging agent to select the proper next question.

The adaptive question selection module is therefore realized through the judging agent to take both ability dimension and question difficulty into consideration. Specifically, we separate the question selection into two stages in order to maintain both efficiency and controllability, including:

Candidate Question Retrieval Since directly asking an agent to choose from the entire remaining pool $\mathcal{Q}/\hat{\mathcal{Q}}$ is impractical, we propose a semantic-aware retrieval mechanism to construct a small candidate set $\mathcal{C}^*$. We aim to select candidate questions that are both appropriate in difficulty and semantically distinct from those previously attempted in $\hat{\mathcal{Q}}$. Firstly, based on the IRT model, we calculate the probability p of the target model M correctly answering each question $x_i \in \mathcal{Q}/\hat{\mathcal{Q}}$ (the detailed formula is provided in Appendix B.1.1). Then we derive a candidate set $\mathcal{C}$ by excluding questions that are either too hard or too easy for the model:

$$\mathcal{C} = \{x_i \in \mathcal{Q}/\hat{\mathcal{Q}} \mid p_{\min} \leq p(x_i) \leq p_{\max}\} \quad (1)$$

Subsequently, to encourage semantic diversity, we apply a max-min retrieval strategy. For each candidate $x \in \mathcal{C}$, we compute its distance to the previously selected question set $\hat{\mathcal{Q}}$ and retain the top-k (k=5) questions with the maximum distance:

$$\mathcal{C}^* = \left\{ x^* = \arg\max_{x \in \mathcal{C}} \min_{x' \in \hat{\mathcal{Q}}} \text{dist}(x, x') \right\} \quad (2)$$

Next Question Selection With the retrieved candidates $\mathcal{C}^*$, we leverage the judging agent to select the next question. The agent is provided with (i) the current ability estimate θ, (ii) the memory $\mathcal{E}$ summarizing ability dimensions and evaluation history, and (iii) the multimodal content of the candidates. Conditioned on the context, it is prompted to reason about which candidate will be most informative for further refining the belief over the model's ability, outputs the selected question, and updates the ability taxonomy within its memory $\mathcal{E}$.

This two-stage design enables A^2-Judger to balance item difficulty with diverse coverage on ability dimensions while maintaining efficiency.

3 Experiments

3.1 Experiment Setup

Benchmarks We conduct experiments on four representative benchmarks: MMMU (Yue et al., 2024), SEEDBench (Li et al., 2024b), MMT-Bench (Ying et al., 2024), and AI2D (Kembhavi et al., 2016). AI2D focuses on the domain of diagram understanding, while the others are used for comprehensive evaluation. All methods are compared under a uniform data budget of 5% (results with other budgets are available in Appendix C.4).

Evaluation Metrics Following the objectives of efficient benchmarking proposed by Perlitz et al. (2023), we design two metrics to evaluate the reliability of the results of efficient evaluation methods. The first metric is **ranking accuracy** $\rho = 1 - \frac{2*\#\text{ Inversions}}{n*(n-1)}$, representing how well an efficient evaluation method preserves the pairwise order of models compared to the full-dataset evaluation. The second metric is **ranking stability**. Given a set of rankings $\{r_1, \ldots, r_K\}$ obtained from K independent trials, the stability score S is calculated as the average pairwise ranking consistency across different runs: $S = \frac{2}{K(K-1)} \sum_{1 \le i < j \le K} \rho(r_i, r_j)$.

Implementation Details A^2-Judger utilizes Qwen2.5-VL-7B (Bai et al., 2025) as its default engine. The impact of the base model is analyzed in Section 3.4. A^2-Judger is implemented in two configurations: A^2-Judger performs agent-based ability decomposition by default, while A^2-Judger$_H$ adopts human-annotated ability taxonomies. We select 17 representative MLLMs for evaluation, while collecting offline responses from 60 additional models for item difficulty estimation as described in Section 2.2. The detailed list of models is provided in Appendix B.2.3. Each experiment is repeated $K = 5$ times, with mean ranking accuracy and ranking stability reported.

Baselines Besides the random baseline, we consider the following methods for comparison, categorized based on the criteria for question selection: (1) **Stratified sampling** based on capability dimensions (via annotated taxonomies or CLIP embedding clusters); (2) **Difficulty-driven methods**, specifically IRT-Greedy (Zhuang et al., 2023) for minimizing an IRT metric (the gap between model capability and question difficulty) in each step and the IRT-clustering approach from tinyBenchmark (Polo et al., 2024); and (3) **Hybrid strategies**, including Stratified/Clustering IRT-G, which perform IRT-Greedy selection within each question group, and a "Hybrid Greedy" strategy based on an objective integrating both embedding distances between questions and the IRT metric. Appendix B.2 provides more details on the implementation of A^2-Judger and other baselines.

3.2 Main Results

Reliability of A^2-Judger As presented in Table 1, the AutoJudger framework provides a systematic basis for organizing and comparing various methods, yielding three key observations: **(1) Both semantic-based ability decomposition and item difficulty are crucial and complementary factors for question selection**. Notably, methods utilizing either factor outperform the random baseline, while those integrating both factors surpass single-factor approaches. **(2) A^2-Judger demonstrates robust generalization capabilities, significantly outperforming other methods** whether using human-defined taxonomies or agent-decomposed dimensions, please see Section 3.3 for further analysis of agent-generated taxonomies. **(3) A^2-Judger exhibits substantially higher stability** than other models—a prerequisite for practical efficient benchmarking. In summary, **A^2-Judger is the most stable and reliable method.**

Method	Ability Decomp.	Ability Estimation	Selection Criteria	Ranking Accuracy (↑) AI2D	MMMU	MMT	SEED	Avg.	Ranking Stability (↑) AI2D	MMMU	MMT	SEED	Avg.
Random	-	Accuracy	Random	91.6	80.1	86.6	88.8	86.8	89.0	72.6	81.0	83.5	81.5
Stratified	Human	Accuracy	Sem	93.5	82.6	85.3	89.0	87.6	91.7	76.2	80.1	86.2	83.5
Cluster	Embed	Accuracy	Sem	93.2	77.9	87.6	91.8	87.6	91.6	73.7	87.4	90.8	85.9
IRT-Greedy	-	IRT	Diff	89.7	81.6	88.2	91.9	87.9	–	–	–	–	–
tinyBenchmarks	-	IRT	Diff	92.9	78.7	87.5	90.9	87.5	91.2	74.4	81.8	85.7	83.3
Clustering IRT-G	Embed	IRT	Diff&Sem	91.0	84.9	87.5	91.8	88.8	92.4	83.5	86.6	91.9	88.6
Stratified IRT-G	Human	IRT	Diff&Sem	93.5	83.8	87.2	91.2	88.9	95.0	82.4	89.9	94.9	90.6
Hybrid-Greedy	Embed	IRT	Diff&Sem	93.4	85.3	90.4	91.2	90.1	–	–	–	–	–
A^2-Judger	A^2-Judger	IRT	Diff&Sem	95.1	87.5	**91.2**	**92.4**	**91.5**	**98.8**	96.0	96.9	**97.1**	97.2
A^2-Judger$_H$	Human	IRT	Diff&Sem	**95.4**	**88.7**	**91.2**	90.7	**91.5**	98.7	**98.4**	**98.4**	95.1	**97.7**

Table 1: **Performance of efficient evaluation methods under a 5% data budget**. Following the general framework of AutoJudger, we distinguish the components of different methods, where 'Diff' and 'Sem' denote question difficulty and semantics, respectively. The **best** and second-best results are highlighted in **bold** and underlined.

Evaluation Method	Evaluated Model Size 3B	7B	72B
GPU Seconds (Per Sample)			
Full-dataset Evaluation	4.52	6.83	781.60
A^2-Judger (7B)	10.11	12.42	787.19
A^2-Judger (3B)	8.18	10.49	785.26
Total Monetary Cost (USD)			
Full-dataset Evaluation	2.32	3.50	400.96
A^2-Judger (7B)	0.26	0.32	20.15
Cost Reduction	**8.92**×	**10.94**×	**19.90**×
A^2-Judger (3B)	0.21	0.27	20.10
Cost Reduction	**11.05**×	**12.97**×	**19.95**×

Table 2: **Empirical evaluation efficiency on MMT**.

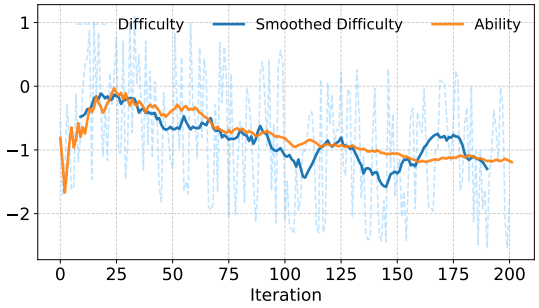


Figure 3: **Evolution of the estimated ability and selected question difficulty during the evaluation of MiniCPM-V2 on MMMU.** "Smoothed difficulty" is calculated through moving average (window size=20).

Efficiency of A^2-Judger Results in Table 2 intuitively illustrate the reduction in evaluation costs achieved by A^2-Judger. Further details on the estimation method can be found in Appendix F.

- **The worst case (Evaluating a 3B model)**: although calling the agent increases the computational cost per sample, A^2-Judger reduces total costs by 9× because it requires only 5% data to achieve a reliable evaluation.
- **For larger models**: For each sample, the impact of the additional agent invocation cost diminishes as the size of the evaluated model increases, becoming almost negligible when evaluating 72B models. A^2-Judger achieves an 11× and 20× reduction in total costs when evaluating 7B and 72B models, respectively. Furthermore, A^2-Judger are more efficient when powered by a 3B backbone, which continues to provide reliable evaluations (see analysis in Section 3.4).

3.3 Interpretability of A^2-Judger

In this section, we analyze the soundness of A^2-Judger's dynamic question selection process by addressing the following two questions:

Q1: Can A^2-Judger perform reasonable ability decomposition? Table 3 compares the capability taxonomies derived from human annotation versus agent analysis. Our findings are as follows: **(1) Annotation granularity varies significantly across datasets**, being either overly fine-grained (MMT) or too coarse (SEED, MMMU); nonetheless, A^2-Judger$_H$ consistently covers a diverse range of dimensions within a 5% data budget. **(2) The judging agent of A^2-Judger effectively balances these dimensions** by refining or consolidating them. For instance, it sub-divides the 9 dimensions in SEED into 24 sub-tasks (yielding a 1.7% improvement in ranking accuracy) and streamlines the 162 dimensions of MMT into 26 (without compromising ranking accuracy). **(3) The capability dimensions identified by the agent are semantically consistent with human labels**; for example, both focus on academic disciplines in MMMU, diagram types in AI2D, and task categories in SEED and MMT.

Q2: Does A^2-Judger strike a balance between question difficulty and dimensional diversity? (1) As illustrated in Figure 3, A^2-Judger adaptively selects questions whose difficulty matches the real-

Benchmark	# of ability dimensions			Typical Category	
	GT	A^2-J$_H$	A^2-J	GT Taxonomy Annotated by Human	Dimensions Decomposed by A^2-Judger
MMMU	6	6	17	Humanities & Social Science, Business, Art & Design Science, Health & Medicine, Tech & Engineering	History, Literature, Economics Physics, Chemistry, Pathology
MMT	162	88	26	Polygon Localization, Color Recognition Weapon Recognition, Artwork Emotion Recognition Facial Expression Recognition, Building Recognition	Bounding Box Description, Color Recognition Weapon Recognition, Emotion Recognition Animal Identification, Time Series Analysis
AI2D	15	13	12	Moon Phase Equinox, Photosynthesis Respiration Faults Earthquakes, Rock Cycle, Rock Strata	Moon Phases, Cellular Respiration, Life Cycle Earth Science, Food Chain, Rock Formation
SEED	9	9	24	Spatial Relation, Instances Counting Scene Understanding, Instance Identity Instance Interaction, Instance Attributes	Spatial Relation Identification, Counting Weather Identification, Location Identification Animal Identification, Clothing Identification

Table 3: **Coverage of capability dimensions.** GT: Total human-defined dimensions in full dataset; A^2-J: Dimensions captured by A^2-Judger within 5% data. Representative dimensions are listed, with colors (blue, red, green, orange) indicating semantic alignment between manually-defined and A^2-Judger-analyzed taxonomies.

Method	AI2D	MMMU	MMT	SEED
Random	0.741	0.806	0.756	0.757
Stratified	0.740	0.817	0.750	0.753
Cluster	0.737	0.767	0.765	0.752
IRT-Greedy	0.745	0.820	0.742	0.757
Clustering IRT-G	0.737	0.852	0.791	0.756
Stratified IRT-G	0.735	0.822	0.747	0.755
tinyBenchmarks	0.744	0.860	0.678	0.747
Hybrid-Greedy	0.745	0.836	0.760	0.757
A^2-Judger	**0.822**	0.932	0.855	**0.813**
w. GT Taxonomy	**0.822**	**0.935**	**0.857**	**0.813**

Table 4: **Average embedding (Euclidean) distance among questions selected by various methods**.

Method	AI2D	MMMU	MMT	SEED	Avg.
A^2-Judger (7B)	**95.1**	**87.5**	91.2	**92.4**	**91.5**
w/o diff.	94.1	84.6	90.4	87.5	89.2
w/o agent	92.4	86.5	86.2	91.8	89.2
A^2-Judger Driven by Different MLLMs					
Qwen2.5-VL-3B	94.4	86.2	**91.8**	91.9	91.1
InternVL2.5-8B	94.1	**87.5**	90.4	91.9	91.0
GPT-4o-mini	–	–	90.4	–	–

Table 5: **Ablation study**. Experiments for the GPT-4o-mini variant are limited to MMT to conserve costs.

time ability of the evaluated model, thereby maximizing difficulty-based information gain. (2) We encode questions with CLIP ViT-B/32 (Radford et al., 2021) and characterize semantic diversity based on the average embedding distance between the selected items. As shown in Table 4, A^2-Judger achieves the most expansive semantic coverage. (3) Beyond quantitative analysis, Appendix E presents qualitative examples that demonstrate A^2-Judger's decision process of jointly optimizing for item difficulty and dimensional diversity.

3.4 Ablation Study

Moreover, we conduct experiments to validate the effectiveness of core designs within A^2-Judger.

Impact of Foundation Model We first explore driving A^2-Judger with a 3B backbone. According to Table 5, a small model is capable of reliable question selection, outperforming all baselines listed in Table 1. This suggests room for further efficiency gains. Additionally, the consistent performance across diverse architectures (InternVL2.5 and GPT-4o) highlights the backbone-agnostic nature and generalization capabilities of A^2-Judger.

Necessity of Question Difficulty Estimation To investigate the role of item difficulty in question selection and the impact of different difficulty estimation methods, we compare 4 settings in MMMU: (i) *IRT-estimated difficulty* (A^2-Judger), (ii) *human-annotated difficulty* (3 levels), (iii) *GPT-4o-mini-annotated difficulty* (5 levels), and (iv) a baseline *excluding difficulty information*.

As illustrated in Figure 4, we observe that: (1) The IRT-estimated difficulty in A^2-Judger consistently outperforms the other configurations. (2) Human-defined difficulty is not suitable for model evaluation and may even underperform the variant without difficulty information, suggesting that item difficulty is best captured through a model-centric perspective; (3) Difficulty measured by GPT-4o-mini offers benefits, serving as a feasible solution for new benchmarks that lack sufficient historical response records. Additionally, we discuss alternative approaches in Appendix C.1, such as utilizing a small set of offline response logs to help adapt A^2-Judger to various scenarios.

Necessity of Agent-based Question Selection To quantify the performance gains provided by the agent-driven question selection strategy relative to statistics-based baselines, we conduct an ablation study. In this case, candidate questions are filtered

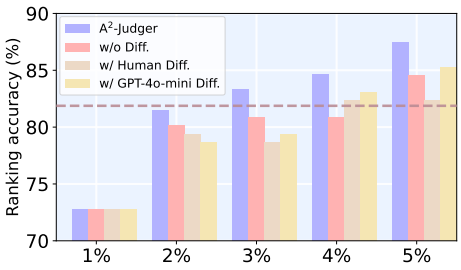


Figure 4: **Performance of A^2-Judger on MMMU under three distinct difficulty settings.** The dashed line is the average accuracy of baselines with 5% data.

using Equation 1, and a weighted sampling mechanism is performed based on the proximity between question difficulty and the model's estimated ability. As shown in Table 5, removing the agent leads to a significant performance drop in four datasets. This indicates the effectiveness of the agent-based strategy in capturing more complex logic than statistical metrics and adapting to diverse scenarios.

In Appendix C, we further demonstrate the robustness of A^2-Judger against various noise and its generalizability across different configurations, highlighting its reliability and practical utility.

4 Related Works

Research on data-efficient evaluation has two directions: **Active testing** starts from zero or a small number of samples, aiming to select as few instances as possible for annotating an effective test set; **Efficient benchmarking**, on the other hand, seeks to exploit existing benchmarks to construct a representative subset for evaluation. Our AutoJudger falls into the latter paradigm.

Active Testing Pioneering attempts primarily follow the active testing setup. Inspired by active learning (Settles, 2009), early work mainly focuses on specific visual cognition tasks, employing various stratified sampling (Bennett and Carvalho, 2010; Kumar and Raj, 2018), Bayesian estimation (Ji et al., 2021), or information gain estimation (Nguyen et al., 2018; Kossen et al., 2021) methods for selection. However, it is challenging to extend such methods to the evaluation of (multimodal) LLMs, because these foundation models are not confined to specific tasks, making it impossible to define the nearly infinite query space and perform effective sampling within it.

Efficient Benchmarking This concept is first introduced by Perlitz et al. (2023) with objective to maintain consistency with full-set evaluation w.r.t. model rankings and scores. Following the setup, we define the corresponding metrics in Section 3.1. Existing efficient benchmarking methods perform question selection based on specific perspectives. The first category of methods employs stratified sampling according to a pre-defined task/scene taxonomy (Perlitz et al., 2023). The second category aims to model the relationship between model performance and question distribution through various aspects: Vivek et al. (2023) reveal a significant correlation in confidence scores predicted by different models to correct answers, and accordingly perform clustering and sampling of questions; other studies mainly rely on the Item Response Theory (IRT) (Lord, 2012; An and Yung, 2014) to model the relationship between question difficulty and model ability. For instance, Polo et al. (2024) selects representative questions by clustering them based on difficulty, while Zhuang et al. (2023) develop an iterative, dynamic question selection strategy based on IRT, which is further extended in subsequent research and supported with theoretical guarantees for its reliability (Zhuang et al., 2025).

Both categories of methods are confined to a specific perspective with its limitations: the former relies heavily on expert-defined capability dimensions, while the latter lacks consideration of multi-dimensional abilities. In contrast, the AutoJudger framework we proposed enables generalization across diverse scenarios, dynamically constructing a capability hierarchy during the evaluation process and integrating it with question difficulty to comprehensively select questions that offer the highest information gain for the current assessment.

5 Conclusions

In this work, we propose AutoJudger, an adaptive evaluation framework to mitigate the escalating computational costs of benchmarking MLLMs. Grounded in cognitive structuralism, our framework transforms evaluation into a dynamic interview via ability decomposition, ability estimation, and adaptive question selection. To address the deficiencies of existing efficient evaluation methods, we introduce A^2-Judger, an agentic instantiation of AutoJudger. Notably, A^2-Judger dispenses with manual ability decomposition through dynamic construction of ability taxonomies during evaluation. Experiments demonstrate that A^2-Judger can provide reliable and stable evaluation results with a 5% data budget, surpassing all baselines.

Limitations

Despite the promising efficiency and stability of AutoJudger, our work has two primary limitations. First, our evaluation is currently centered on discriminative tasks (e.g., multiple-choice questions), where response correctness is well-defined and binary. Extending the framework to open-ended generation tasks, where evaluating response quality is inherently subjective and necessitates complex reward modeling, remains an important direction for future research. Second, the scope of our experiments is confined to image-text benchmarks. The applicability of A^2-Judger to a broader range of modalities, such as video or audio, remains underexplored. These domains introduce temporal dimensions containing richer and more complex semantic-temporal information. Therefore, designing improved mechanisms for ability decomposition, ability estimation, and adaptive question selection to accommodate these factors represents a valuable avenue for exploration. By addressing these limitations, we can enhance the scalability of our method, further advancing the development of efficient benchmarking.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, and 1 others. 2024. Pixtral 12b. *arXiv preprint arXiv:2410.07073*.

Xinming An and Yiu-Fai Yung. 2014. Item response theory: What it is and how you can use the irt procedure to apply it. *SAS Institute Inc*, 10(4):364–2014.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *Preprint*, arXiv:2308.12966.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Paul N. Bennett and Vitor R. Carvalho. 2010. Online stratified sampling: evaluating classifiers at web-scale. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 1581–1584, New York, NY, USA. Association for Computing Machinery.

Lucas Beyer*, Andreas Steiner*, André Susano Pinto*, Alexander Kolesnikov*, Xiao Wang*, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, and 16 others. 2024. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*.

Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. 2017. Pyro: Deep universal probabilistic programming. *arXiv preprint arXiv:1701.02434*.

Denny Borsboom. 2005. *Measuring the mind: Conceptual issues in contemporary psychometrics*. Cambridge University Press.

Li Cai, Kilchan Choi, Mark Hansen, and Lauren Harrell. 2016. Item response theory. *Annual Review of Statistics and Its Application*, 3(1):297–321.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. 2025. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024a. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, and 1 others. 2024b. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

China Mobile Communications Corporation. 2024. https://github.com/jiutiancv/JT-VL-Chat.

Wenliang Dai, Nayeon Lee, Boxin Wang, Zhuolin Yang, Zihan Liu, Jon Barker, Tuomas Rintamaki, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nvlm:

Open frontier-class multimodal llms. *arXiv preprint arXiv:2409.11402*.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, and 31 others. 2024a. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. *Preprint*, arXiv:2409.17146.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, and 1 others. 2024b. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*.

Mucong Ding, Chenghao Deng, Jocelyn Choo, Zichu Wu, Aakriti Agrawal, Avi Schwarzschild, Tianyi Zhou, Tom Goldstein, John Langford, Animashree Anandkumar, and 1 others. 2024. Easy2hard-bench: Standardized difficulty labels for profiling llm performance and generalization. *Advances in Neural Information Processing Systems*, 37:44323–44365.

Khang T. Doan, Bao G. Huynh, Dung T. Hoang, Thuc D. Pham, Nhat H. Pham, Quan T. M. Nguyen, Bang Q. Vo, and Suong N. Hoang. 2024. Vintern-1b: An efficient multimodal large language model for vietnamese. *Preprint*, arXiv:2408.12480.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, and 4 others. 2024. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *Preprint*, arXiv:2401.16420.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, and 1 others. 2024. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 11198–11201.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and 1 others. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Shaikat Galib, Shanshan Wang, Guanshuo Xu, Pascal Pfeiffer, Ryan Chesler, Mark Landry, and Sri Satish Ambati. 2024. H2ovl-mississippi vision language models technical report. *arXiv preprint arXiv:2410.13611*.

Wentao Ge, Shunian Chen, Hardy Chen, Nuo Chen, Junying Chen, Zhihong Chen, Wenya Xie, Shuo Yan, ChenghaoZhu ChenghaoZhu, Ziyue Lin, and 1 others. 2025. Mllm-bench: evaluating multimodal llms with per-sample criteria. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4951–4974.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Shuhao Gu, Jialing Zhang, Siyuan Zhou, Kevin Yu, Zhaohu Xing, Liangdong Wang, Zhou Cao, Jintao Jia, Zhuoyi Zhang, Yixuan Wang, Zhenchong Hu, Bo-Wen Zhang, Jijie Li, Dong Liang, Yingli Zhao, Yulong Ao, Yaoqi Liu, Fangxiang Feng, and Guang Liu. 2024. Infinity-mm: Scaling multimodal performance with large-scale and high-quality instruction data. *Preprint*, arXiv:2410.18558.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. 2024. Efficient multimodal learning from data-centric perspective. *Preprint*, arXiv:2402.11530.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Disi Ji, IV RobertL.Logan, Padhraic Smyth, and Mark Steyvers. 2021. Active bayesian assessment of black-box classifiers. In *AAAI Conference on Artificial Intelligence*.

Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. 2024a. Mantis: Interleaved multi-image instruction tuning. *Preprint*, arXiv:2405.01483.

Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max W.F. Ku, Qian Liu, and Wenhu Chen. 2024b. Mantis: Interleaved multi-image instruction tuning. *Transactions on Machine Learning Research*, 2024.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Diederik P Kingma and Max Welling. 2022. Auto-encoding variational bayes. *Preprint*, arXiv:1312.6114.

Vikhyat Korrapati. 2024. Moondream2: A vision-language model. `https://huggingface.co/vikhyatk/moondream2`.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. 2021. Active testing: Sample-efficient model evaluation. *Preprint*, arXiv:2103.05331.

Anurag Kumar and Bhiksha Raj. 2018. Classifier risk estimation under limited labeling resources. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part I*, page 3–15, Berlin, Heidelberg. Springer-Verlag.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024. What matters when building vision-language models? *Advances in Neural Information Processing Systems*, 37:87874–87907.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. Llava-onevision: Easy visual task transfer. *Preprint*, arXiv:2408.03326.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024b. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13299–13308.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*.

Zejun Li, Ruipu Luo, Jiwen Zhang, Minghui Qiu, Xuanjing Huang, and Zhongyu Wei. 2024c. Vocot: Unleashing visually grounded multi-step reasoning in large multi-modal models. *arXiv preprint arXiv:2405.16919*.

Zejun Li, Ye Wang, Mengfei Du, Qingwen Liu, Binhao Wu, Jiwen Zhang, Chengxing Zhou, Zhihao Fan, Jie Fu, Jingjing Chen, and 1 others. 2024d. Reform-eval: Evaluating large vision language models via unified re-formulation of task-oriented benchmarks. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1971–1980.

Zejun Li, Jiwen Zhang, Dianyi Wang, Ye Wang, Xuanjing Huang, and Zhongyu Wei. 2024e. Continuous or discrete, that is the question: A survey on large multimodal models from the perspective of input-output space extension.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2024f. Monkey: Image resolution and text label are important things for large multi-modal models. *Preprint*, arXiv:2311.06607.

Zongxia Li, Xiyang Wu, Hongyang Du, Fuxiao Liu, Huy Nghiem, and Guangyao Shi. 2025. A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024b. Improved baselines with visual instruction tuning. *Preprint*, arXiv:2310.03744.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024c. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024d. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Yuan Liu, Zhongyin Zhao, Ziyuan Zhuang, Le Tian, Xiao Zhou, and Jie Zhou. 2024e. Points: Improving your vision-language model with affordable strategies. *Preprint*, arXiv:2409.04828.

Frederic M Lord. 2012. *Applications of item response theory to practical testing problems*. Routledge.

DataCanvas Ltd. 2024. Mmalaya2. `https://huggingface.co/DataCanvas/MMAlaya2`.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. 2024a. Deepseek-vl: Towards real-world vision-language understanding. *Preprint*, arXiv:2403.05525.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. 2024b. Deepseek-vl: Towards real-world vision-language understanding. *Preprint*, arXiv:2403.05525.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. 2024c. Ovis: Structural embedding alignment for multimodal large language model. *Preprint*, arXiv:2405.20797.

Xudong Lu, Yinghao Chen, Cheng Chen, Hui Tan, Boheng Chen, Yina Xie, Rui Hu, Guanxin Tan, Renshou Wu, Yan Hu, Yi Zeng, Lei Wu, Liuyang Bian, Zhaoxiong Wang, Long Liu, Yanzhou Yang, Han Xiao, Aojun Zhou, Yafei Wen, and 3 others. 2024d. Bluelm-v-3b: Algorithm and system co-design for multimodal large language models on mobile devices. *Preprint*, arXiv:2411.10640.

Ling Luo, Jinzhong Ning, Yingwen Zhao, Zhijun Wang, Zeyuan Ding, Peng Chen, Weiru Fu, Qinyu Han, Guangtao Xu, Yunzhi Qiu, and 1 others. 2024. Taiyi: a bilingual fine-tuned large language model for diverse biomedical tasks. *Journal of the American Medical Informatics Association*, 31(9):1865–1874.

Phuc Nguyen, Deva Ramanan, and Charless Fowlkes. 2018. Active testing: An efficient and robust framework for estimating accuracy. *Preprint*, arXiv:1807.00493.

OpenAI. 2023. Gpt-4v-system-card. https://openai.com/index/gpt-4v-system-card.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NeurIPS Workshop*.

PCIResearch. 2023. Transcore-m: Multimodal foundation model for transportation research. https://github.com/PCIResearch/TransCore-M.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. 2023. Efficient benchmarking of language models. *arXiv preprint arXiv:2308.11696*.

Jean Piaget. 2015. *Structuralism (psychology revivals)*. Psychology Press.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*.

Qihoo360 AI Lab. 2024. qihoo360/360vl-70b: An open-source large vision-language model based on llama3-70b. https://huggingface.co/qihoo360/360VL-70B. Accessed: 2025-MM-DD.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Georg Rasch. 1993. *Probabilistic models for some intelligence and attainment tests.* ERIC.

Burr Settles. 2009. Active learning literature survey.

Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, Bryan Catanzaro, Andrew Tao, Jan Kautz, Zhiding Yu, and Guilin Liu. 2024. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *arXiv:2408.15998*.

StepFun. 2024. Step-1v. https://platform.stepfun.com/docs/llm/vision.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Technology Innovation Institute (TII). 2024. tiiuae/falcon-11b-vlm: A vision-language model based on falcon-11b. https://huggingface.co/tiiuae/falcon-11B-vlm.

Silvia Terragni, Hoang Cuong, Joachim Daiber, Pallavi Gudipati, and Pablo N Mendes. 2024. Evaluating cost-accuracy trade-offs in multimodal search relevance judgements. *arXiv preprint arXiv:2410.19974*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jonathan Y Tsou. 2006. Genetic epistemology and piaget's philosophy of science: Piaget vs. kuhn on scientific progress. *Theory & Psychology*, 16(2):203–224.

Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R Bowman. 2021. Comparing test sets with item response theory. *arXiv preprint arXiv:2106.00840*.

vikhyatk. 2024. Moondream1. https://huggingface.co/vikhyatk.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. 2023. Anchor points: Benchmarking models with much fewer examples. *arXiv preprint arXiv:2309.08638*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and 1 others. 2024. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*.

Siyu Xu, Yunke Wang, Daochang Liu, Bo Du, and Chang Xu. 2025. Collageprompt: A benchmark for budget-friendly visual recognition with gpt-4v. In *Find-*

ings of the Association for Computational Linguistics: NAACL 2025, pages 6396–6418.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403.

Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, and 1 others. 2024. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi. *arXiv preprint arXiv:2404.16006*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Yi-Fan Zhang, Qingsong Wen, Chaoyou Fu, Xue Wang, Zhang Zhang, Liang Wang, and Rong Jin. 2024a. Beyond llava-hd: Diving into high-resolution large multimodal models. *arXiv preprint arXiv:2406.08487*.

Yi-Fan Zhang, Qingsong Wen, Chaoyou Fu, Xue Wang, Zhang Zhang, Liang Wang, and Rong Jin. 2024b. Beyond llava-hd: Diving into high-resolution large multimodal models. *Preprint*, arXiv:2406.08487.

Yan Zhuang, Qi Liu, Yuting Ning, Weizhe Huang, Rui Lv, Zhenya Huang, Guanhao Zhao, Zheng Zhang, Qingyang Mao, Shijin Wang, and 1 others. 2023. Efficiently measuring the cognitive ability of llms: An adaptive testing perspective. *CoRR*.

Yan Zhuang, Junhao Yu, Qi Liu, Yuxuan Sun, Jiatong Li, Zhenya Huang, and Enhong Chen. 2025. Efficient benchmarking via bias-bounded subset selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

A Details of Evaluation Benchmarks

	AI2D	MMMU	MMT	SEED
Split	*TEST*	*DEV&VAL*	*VAL*	*IMG*
# Samples	3088	1050	3127	1423
Category	Semantic Category	Discipline	Subtask	Evaluation Dimension

Table 6: The split(s), sample sizes, and categories we use of the four benchmarks in our experiments.

Table 6 introduces the multimodal evaluation benchmarks utilized in this work, where 'Category' refers to the human-defined capability taxonomy associated with each dataset.

B Supplementary Implementation Details

B.1 Ability-Difficulty Estimation System

In this section, we elaborate on the system that is used to estimate the model abilities and question difficulty, supplementing the introduction in Section 2.2. Appendix B.1.1 describes the IRT model we utilize in this paper, while Appendix B.1.2 introduces how to estimate the real-time model ability during the evaluation process.

B.1.1 Details of Rasch Model (IRT) Fitting

Modeling with IRT We adopt a logistic IRT model, also known as the Rasch model (Rasch, 1993), which defines the probability of a correct response r_{ij} by model m_j on question q_i as:

$$p\left(r_{ij}=1 \mid a_j, d_i\right)=\frac{1}{1+\exp(-(a_j-d_i))} \quad (3)$$

where a_j represents the latent ability of the model m_j, and d_i denotes the difficulty of the question q_i. Intuitively, a model is more likely to succeed on questions with difficulty levels lower than its ability level.

Variational Bayesian Framework We use variational inference to approximate the posterior distribution over model abilities and question difficulties. Specifically, we assume a fully factorized variational distribution (Ding et al., 2024):

$$q(a, D)=\prod_j q(a_j) \prod_i q(d_i) \quad (4)$$

Each latent variable is modeled as a Gaussian:

$$\begin{aligned} q(a_j) &= \mathcal{N}(\mu_{a_j}, \sigma_{a_j}^2) \\ q(d_i) &= \mathcal{N}(\mu_{d_i}, \sigma_{d_i}^2) \end{aligned} \quad (5)$$

The optimization target is the evidence lower bound (ELBO):

$$\begin{aligned} \mathcal{L}_{\text{ELBO}} =& \mathbb{E}_{q(a,D)}[\log p(r \mid a, D)] \\ & - \mathrm{KL}(q(a, D) \parallel p(a, D)) \end{aligned} \quad (6)$$

We adopt standard Gaussian priors: $p(a_j) = \mathcal{N}(0,1)$ and $p(d_i) = \mathcal{N}(0, 10^3)$, which yield closed-form KL divergences.

Optimization We optimize the ELBO using stochastic gradient descent. Gradients are estimated via the reparameterization trick:

$$\begin{aligned} a_j &= \mu_{a_j} + \sigma_{a_j} \cdot \epsilon_j, \quad \epsilon_j \sim \mathcal{N}(0,1) \\ d_i &= \mu_{d_i} + \sigma_{d_i} \cdot \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0,1) \end{aligned} \quad (7)$$

This leads to efficient and low-variance updates for the variational parameters μ and σ.

Implementation Setting We implement the model using PyTorch (Paszke et al., 2017) and Pyro (Bingham et al., 2017). The variational distributions over model abilities and question difficulties are initialized with zero mean and large variance. Specifically, the ability parameters a_j are initialized with $\mu_{a_j} = 0$, $\sigma_{a_j} = 1$, while the difficulty parameters d_i are initialized with $\mu_{d_i} = 0$, $\sigma_{d_i} = 10^3$, corresponding to vague priors that reflect minimal prior knowledge.

We optimize the ELBO (Kingma and Welling, 2022) using the Adam optimizer (Kingma and Ba, 2017) with a learning rate of 0.1 for 3,200 steps, using mini-batches sampled from the response matrix $\{r_{ij}\}$. Training terminates when the relative change in ELBO falls below 1×10^{-4} within a moving window. During inference, we use the variational mean μ_{d_i} as the point estimate of question difficulty.

B.1.2 Details of Model Ability Estimation

To estimate the model ability a_j based on its responses to a subset of questions, we employ a binary search algorithm grounded in the one-parameter logistic IRT (Rasch) model. Specifically, we solve the following maximum likelihood estimation problem:

$$\max_{a_j} \sum_{i \in Q'} \log p(r_{ij} \mid a_j, d_i), \quad (8)$$

where $p(r_{ij} \mid a_j, d_i)$ is defined in Equation 3. Since the log likelihood is a monotonic function with respect to a_j, we perform binary search within a bounded interval $[-30, 30]$, iteratively updating the estimate until convergence. The stopping criterion is based on a fixed threshold of 10^{-5} for either the log-likelihood difference or the change in a_j.

This procedure enables efficient and stable estimation of real-time model ability during evaluation, while keeping the question difficulties $\{d_i\}$ fixed.

B.2 Implementation Details of Efficient Benchmarking Methods

B.2.1 Baselines

Random methods:

- *Random Sampling (Random)*: We uniformly sample $\delta * |Q|$ questions from the complete evaluation benchmark Q without replacement.

Methods based on ability decomposition:

- *Stratified Random Sampling (Stratified Random)* (Perlitz et al., 2023): We partition the question pool based on provided category labels and draw approximately equal numbers of questions from each partition. We ensure that the maximum size difference between any two categories is no greater than one. Sampling is performed independently per category without replacement.

- *Cluster-Based Sampling (Cluster)*: Each question is embedded using the CLIP ViT-B/32 encoder (Radford et al., 2021), producing a 512-dimensional representation. Embeddings are L2-normalized before clustering. We apply K-means clustering to partition the question pool. The number of clusters K is set to the number of desired questions, i.e., $K = \delta \cdot |Q|$. One question is selected per cluster, chosen as the one closest to the centroid.

Methods driven by difficulty:

- *Optimal IRT Difficulty Choosing (IRT-Greedy)* (Lord, 2012): We use a one-parameter logistic Item Response Theory model to adaptively select questions based on the model's estimated ability. The ability score is initialized with a simple prior: we assume the model has answered five medium-difficulty questions (difficulty 0) and got 2.5 correct on average. This initialization prevents unstable updates in early iterations. It can be demonstrated that this approach is equivalent to the method of maximizing Fisher information described in (Zhuang et al., 2023). The detailed proof is provided in Appendix B.2.2.

- *tinyBenchmarks* (Polo et al., 2024): Following the original settings, we employ Item Response Theory parameters as the semantic representation of questions. Specifically, we fit a two-parameter multidimensional IRT model where the probability of a model answering a question correctly is determined by the discrimination vector α_i and the difficulty term β_i. We utilize these estimated parameters to form the question embedding E_i. Questions are then sampled by clustering these embeddings to identify anchor points, with weights assigned based on cluster density.

Methods that jointly consider both difficulty and semantic diversity:

- *Stratified Optimal IRT Difficulty Choosing (Stratified IRT-G)* : We partition the question pool based on the provided category labels. At each step, we select the category with the fewest questions and apply an optimal IRT-based difficulty selection strategy.

- *Clustering Optimal IRT Difficulty Choosing (Clustering IRT-G)*: Instead of relying on a predefined taxonomy, we partition the question pool into K latent semantic clusters (e.g., $K = 10$) using the K-Means algorithm on the questions' feature embeddings. To ensure content diversity, we employ a round-robin scheduling strategy to cycle through clusters.

- *Dual Distance Minimization (Hybrid-Greddy)* : We define the difficulty-ability distance as the absolute difference between the question difficulty and the model ability, which corresponds to the optimization objective of the optimal IRT difficulty selection strategy. Additionally, the semantic distance is defined as the minimum distance between the embedding vector of a candidate question and those of all previously tested questions. Given that both the embedding vectors and difficulty values are normalized, we directly sum two metrics to form the minimization objective for determining the question selection at each step.

B.2.2 The Equality between Two Difficulty-Driven Selection Methods

We provide the detailed proof that maximizing Fisher information is equivalent to selecting a question with difficulty d closest to ability a (i.e., minimize $|d-a|$). The probability of a correct response in the 1PL model is $P(a) = \frac{1}{1+e^{d-a}}$. First, we compute the derivative of $P(a)$ with respect to a:

$$P'(a) = \frac{\partial}{\partial a}(1+e^{d-a})^{-1} = \frac{e^{d-a}}{(1+e^{d-a})^2}. \quad (9)$$

The Fisher information $I(a)$ is defined as $I(a) = \frac{[P'(a)]^2}{P(a)(1-P(a))}$. Substituting the expressions, we obtain:

$$\begin{aligned} I(a) &= \frac{\left(\frac{e^{d-a}}{(1+e^{d-a})^2}\right)^2}{\frac{1}{1+e^{d-a}} \cdot \frac{e^{d-a}}{1+e^{d-a}}} = \frac{\frac{(e^{d-a})^2}{(1+e^{d-a})^4}}{\frac{e^{d-a}}{(1+e^{d-a})^2}} \\ &= \frac{e^{d-a}}{(1+e^{d-a})^2}. \end{aligned} \quad (10)$$

To find the maximum of $I(\theta)$, we divide both the numerator and the denominator by $e^{d-\theta}$:

$$\begin{aligned} I(a) &= \frac{1}{e^{-(d-a)} + 2 + e^{d-a}} \\ &= \frac{1}{2 + e^{|d-a|} + e^{-|d-a|}}. \end{aligned} \quad (11)$$

Consequently, $I(a)$ is monotonically decreasing with respect to $|d-a|$. Therefore, the strategy of maximizing Fisher information is equivalent to selecting the item with difficulty closest to the current examinee's ability.

B.2.3 Training and Test Models

To ensure a representative and balanced evaluation, we partition the models based on parameter scale, as the model capability is generally observed to improve with increasing parameter size. Accordingly, we divide the models into four groups: $< 5B$, $< 9B$, $< 16B$, and $\geq 16B$ (including proprietary models). From each group, we randomly sample 20% of the models as test models, the remaining 80% are used as training models to collect offline responses[1], which are utilized for question difficulty estimation. This stratified selection strategy ensures that AutoJudger is evaluated across a wide spectrum in terms of model abilities. The complete list of the 60 training models used for IRT-based question difficulty assessment is provided in Table 7, and the remaining 17 models evaluated are listed in Table 8.

[1]Offline results are collected from VLMEvalKit (Duan et al., 2024): https://github.com/open-compass/VLMEvalKit.

Models	Open-source	# Params (B)	Date
InternVL2-1B (Chen et al., 2024c)	Yes	0.9	2024.11
llava-onevision-qwen2-0.5B-ov (Li et al., 2024a)	Yes	0.9	2024.07
llava-onevision-qwen2-0.5B-si (Li et al., 2024a)	Yes	0.9	2024.07
h2ovl-mississippi-1B (Galib et al., 2024)	Yes	0.8	2024.01
NVLM (Dai et al., 2024)	Yes	79.4	2024.09
Qwen2-VL-72B-Instruct (Wang et al., 2024)	Yes	72	2024.08
360VL-70B (Qihoo360 AI Lab, 2024)	Yes	71	2024.04
InternVL2-40B (Chen et al., 2024c)	Yes	40.1	2024.06
InternVL-Chat-V1-5 (Chen et al., 2024c)	Yes	25.5	2024.03
InternVL2-26B (Chen et al., 2024c)	Yes	25.5	2024.11
MMAlaya2 (Ltd., 2024)	Yes	25.5	2024.08
Eagle-X5-13B (Shi et al., 2024)	Yes	15.4	2024.08
Slime-13B (Zhang et al., 2024a)	Yes	13.4	2024.05
TransCore-M (PCIResearch, 2023)	Yes	13.4	2024.03
llava-v1.5-13B (Liu et al., 2024b)	Yes	13	2024.01
Falcon2-VLM-11B (2024)	Yes	11	2024.07
Ovis1.6-Gemma2-9B (Lu et al., 2024c)	Yes	10.2	2024.09
monkey (Li et al., 2024f)	Yes	9.8	2023.11
monkey-chat (Li et al., 2024f)	Yes	9.8	2023.11
POINTS-Yi-1.5-9B-Chat (Liu et al., 2024e)	Yes	9.5	2024.09
Mantis-8B-Fuyu (Jiang et al., 2024b)	Yes	9.4	2024.04
Eagle-X5-7B (Shi et al., 2024)	Yes	9.1	2024.08
Bunny-llama3-8B (He et al., 2024)	Yes	8.5	2024.04
Mantis-8B-siglip-llama3 (Jiang et al., 2024a)	Yes	8.5	2024.04
Mantis-8B-Idefics2 (Jiang et al., 2024a)	Yes	8.4	2024.05
Slime-8B (Zhang et al., 2024b)	Yes	8.4	2024.05
llava-next-llama3 (Liu et al., 2024c)	Yes	8.3	2024.04
POINTS-Qwen-2.5-7B-Chat (Liu et al., 2024e)	Yes	8.3	2024.12
llava-next-interleave-7B (Liu et al., 2024c)	Yes	8.1	2024.06
llava-next-interleave-7B-dpo (Liu et al., 2024c)	Yes	8.1	2024.06
MiniCPM-V-2-6 (Yao et al., 2024)	Yes	8.1	2024.07
InternVL2-8B (Chen et al., 2024b)	Yes	8.1	2024.11
llava-onevision-qwen2-7B-ov (Li et al., 2024a)	Yes	8.0	2024.07
Ovis1.5-Llama3-8B (Lu et al., 2024c)	Yes	8	2024.07
molmo-7B-O-0924 (Deitke et al., 2024a)	Yes	7.7	2024.09
llava-next-mistral-7B (Liu et al., 2024c)	Yes	7.6	2024.03
deepseek-vl-7B (Lu et al., 2024a)	Yes	7.3	2024.02
llava-next-vicuna-7B (Liu et al., 2024c)	Yes	7.1	2024.05
XComposer2 (Dong et al., 2024)	Yes	7	2024.01
llava-v1.5-7B (Liu et al., 2024b)	Yes	7	2024.01
Phi-3-Vision (Abdin et al., 2024)	Yes	4.2	2024.05
InternVL2-4B (Chen et al., 2024b)	Yes	3.7	2024.11
Vintern-3B-beta (Doan et al., 2024)	Yes	3.2	2024.01
BlueLM-V (Lu et al., 2024d)	No	3	2024.11
paligemma-3B-mix-448 (Beyer* et al., 2024)	Yes	2.9	2024.04
InternVL2-2B (Chen et al., 2024b)	Yes	2.2	2024.11
Aquila-VL-2B (Gu et al., 2024)	Yes	2.2	2024.01
deepseek-vl-1.3B (Lu et al., 2024b)	Yes	2.0	2024.02
Moondream1 (vikhyatk, 2024)	Yes	1.9	2024.01
XComposer2-1.8B (Dong et al., 2024)	Yes	1.8	2024.01
Kosmos2 (Peng et al., 2023)	Yes	1.7	2023.06
molmoE-1B-0924 (Deitke et al., 2024b)	Yes	1	2024.09
GPT4V-20240409-HIGH (OpenAI, 2023)	No	-	2024.04
GPT4o (Hurst et al., 2024)	No	-	2024.05
GPT4o-HIGH (Hurst et al., 2024)	No	-	2024.05
GeminiFlash1-5 (Team et al., 2024)	No	-	2024.09
JT-VL-Chat (Corporation, 2024)	No	-	2024.10
Qwen-VL-Max-0809 (Bai et al., 2023)	No	-	2024.08
Qwen-VL-Plus-0809 (Bai et al., 2023)	No	-	2024.08
Taiyi (Luo et al., 2024)	No	-	2023.11

Table 7: **List of 60 training set models used for IRT-based question difficulty assessment.** These models span a range of sizes and include both open-source and proprietary models.

B.2.4 Our Framework: A^2-Judger

The evaluation workflow of A^2-Judger follows Algorithm 1. All questions in each benchmark are first supplemented with estimated difficulty levels. Then, evaluation begins with a standardized initialization, followed by iterative refinement of the question set based on the model's responses. Our code and data are available at https://anonymous.4open.science/r/AutoJudger-anonymous.

Inference Hyperparameters To minimize randomness and ensure the reproducibility of our experimental results, we utilize a near-deterministic

Models	Open-source	# Params (B)	Date
InternVL2-76B (Chen et al., 2024a)	Yes	76.3	2024.06
llava-next-vicuna-13B (Liu et al., 2024c)	Yes	13.4	2024.02
Pixtral-12B (Agrawal et al., 2024)	Yes	12	2024.08
Ovis1.5-Gemma2-9B (Lu et al., 2024c)	Yes	11.4	2024.07
idefics2-8B (Laurençon et al., 2024)	Yes	8.4	2024.03
Mantis-8B-clip-llama3 (Jiang et al., 2024b)	Yes	8.3	2024.01
llava-onevision-qwen2-7B_si (Li et al., 2024a)	Yes	8.0	2024.07
molmo-7B-D-0924 (Deitke et al., 2024a)	Yes	8.0	2024.09
Slime-7B (Zhang et al., 2024a)	Yes	7.1	2024.05
Ovis1.6-Llama3.2-3B (Lu et al., 2024c)	Yes	4.1	2024.01
MiniCPM-V-2 (Yao et al., 2024)	Yes	3.4	2024.11
h2ovl-mississippi-2B (Galib et al., 2024)	Yes	2.1	2024.01
Janus-1.3B (Wu et al., 2024)	Yes	2.1	2024.01
Moondream2 (Korrapati, 2024)	Yes	1.9	2024.02
GPT4o-20240806 (Hurst et al., 2024)	No	-	2024.08
GeminiPro1-5 (Team et al., 2024)	No	-	2024.09
Step1V (StepFun, 2024)	No	-	2024.03

Table 8: **17 test set models evaluated.** These models are disjoint from the training set and representative in terms of diverse types and scales.

decoding strategy for all model inferences. Specifically, we configure the generation hyperparameters by setting the temperature to 1×10^{-6}, top-k to 50, and top-p to 1.0.

Initialization Details In the initialization phase, we take the text of each question as input and use CLIP ViT-B/32 as the encoding model to generate normalized vector representations. We aim to select a diverse and representative set of questions, so we apply k-means clustering with k=10 and select the questions closest (in terms of L2 distance) to each of the resulting cluster centers. To mitigate the instability of k-means, we repeat the clustering process 50 times and choose the set that achieves the highest ranking accuracy on the training set.

Implementation of Ability Decomposition in A^2-Judger To maintain contextual coherence during evaluation, A^2-Judger supports a memory mechanism. Emphasizing long-term statistical awareness, the memory $\mathcal{M}$ accumulates high-level information about previously selected questions and model responses, grouped by categories as a markdown-form table. Regarding ability decomposition, the strategy differs by configuration: for A^2-Judger, since many benchmarks lack predefined labels or contain noisy annotations, ability is decomposed by the MLLM-driven agent based on semantically inferred features, dynamically expanding as new topics emerge (prompts are provided in Appendix G). In contrast, for the A^2-Judger$_H$, we directly utilize the original category labels provided by the benchmarks (as detailed in Appendix A). For each category, the memory tracks statistics including the number of questions, max/min/average difficulty, and overall accuracy. This enables the agent to maintain global awareness of coverage and balance across domains. A representative example of the memory table is shown in Table 9.

Category	**Count**	**Max Diff.**	**Min Diff.**	**Avg Diff.**	**Acc.**
Accounting	5	5.21	-1.02	1.15	0.60
Art History	20	1.01	-5.20	-0.83	0.71
Botany	9	0.45	-5.30	-1.31	0.56
Cell Biology	14	4.90	-2.44	-0.70	0.50

Table 9: An example of the memory table maintained by A^2-Judger.

Benchmark	Methods	Evaluation Data Budget				
		1%	2%	3%	4%	5%
AI2D	IRT_{60}	**86.62**	**91.03**	**93.24**	**94.71**	**95.15**
	IRT_{20}	85.29	90.44	92.65	92.65	94.85
MMMU	IRT_{60}	**77.79**	**81.47**	83.38	**84.71**	**87.50**
	IRT_{20}	**77.79**	80.15	**84.56**	84.56	**87.50**
	GPT-4o-mini	**77.79**	78.68	79.41	83.09	85.29
MMT	IRT_{60}	80.59	**88.82**	88.68	90.59	91.18
	IRT_{20}	**81.62**	86.76	**91.18**	**93.38**	**94.12**
	GPT-4o-mini	75.74	80.88	86.03	89.71	88.97
SEED	IRT_{60}	88.97	**91.62**	**91.76**	**92.21**	**92.35**
	IRT_{20}	**89.71**	90.44	89.71	91.18	88.97

Table 10: **Comparison of Difficulty Annotation Schemes.**

This memory table illustrates a more realistic usage scenario, featuring a broad range of question difficulties and imbalanced category distributions. For example, Art History contains 20 questions spanning a wide difficulty range with relatively high accuracy, while Accounting tends to be more difficult with greater outcome variance. Such statistical tracking allows the agent to identify underrepresented or overly challenging areas, informing more targeted selection in subsequent iterations.

C Further Analysis on Generalization and Stability of A^2-Judger across Different Factors

In this section, we investigate how different factors affect the performance of A^2-Judger, including: different difficulty estimation methods (C.1), different question selection strategies (C.2), different IRT settings (C.3), and different data budgets (C.4).

C.1 Alternative Difficulty Estimation Methods

AutoJudger derives difficulty annotations for benchmark questions by applying IRT to the response logs of 60 existing models. Acknowledging that obtaining such extensive response data may be challenging for newly released benchmarks, we simulate two alternative approaches to address this scenario: annotating difficulty using a reduced set of historical response logs, and leveraging a powerful closed-source model (e.g., GPT-4o-mini) for

Benchmark	Methods	Evaluation Data Budget				
		1%	2%	3%	4%	5%
AI2D	personalized	**86.62**	**91.03**	**93.24**	**94.71**	**95.15**
	simplest	86.03	89.56	89.56	91.91	93.24
MMT	personalized	**80.59**	**88.82**	**88.68**	**90.59**	**91.18**
	simplest	78.82	78.68	82.06	85.15	87.94
SEED	personalized	**88.97**	**91.62**	**91.76**	92.21	92.35
	simplest	85.29	89.56	89.85	**92.65**	**93.97**

Table 11: **Ranking accuracy for top-performing models with personalized and unified question selection strategies.** "personalized" means the questions are selected via Equation 1. "simplest" means the questions are from the evaluation of the lowest-ranked model and fixed for all models.

direct difficulty annotation. The results in Table 10 indicate negligible performance degradation, suggesting that A^2-Judger is not overly sensitive to the precision of difficulty annotations. This low dependency highlights its robust adaptability to benchmarks across diverse scenarios.

C.2 The Impact of Candidate Question Retrieval Strategy

Superiority of Personalized Retrieval As argued in the introduction, we believe each model should be assigned with a personalized evaluation subset since models vary in capability. For instance, evaluating powerful models with too many easy questions may provide limited information. To validate this argument, we conduct an experiment to assess top-performing models (top 50% in terms of average ranks), either with their personalized questions picked by A^2-Judger or simple questions that are selected to evaluate the worst model.

Results are provided in Table 11. Considering the efficiency, SEEDBench is excluded in this experiment due to its large scale. Since AI2D is a relatively easy scenario (minimal variation in difficulty across the questions), the "simplest" strategy demonstrates comparative performance. However, on more complex benchmarks like MMT and MMMU, the personalized approach demonstrates superior performance. Additionally, we observed that the "simplest" strategy lacks stability and does not necessarily improve as the dataset size increases. Generally, by dynamically selecting questions tailored to each model's capability, the proposed A^2-Judger better accommodates varying model strengths and avoids overfitting to the preference of specific models. Therefore, we adopt the personalized strategy to retrieve questions.

Benchmark	Strategy	Evaluation Data Budget				
		1%	2%	3%	4%	5%
AI2D	semantic farthest	**86.62**	**91.03**	**93.24**	**94.71**	**95.15**
	optimal difficulty	86.03	90.44	89.71	89.71	91.91
	random	79.41	89.71	92.65	92.65	93.38
MMMU	semantic farthest	**77.79**	**81.47**	**83.38**	**84.71**	**87.50**
	optimal difficulty	**77.79**	77.94	80.88	83.09	86.76
	random	**77.79**	78.68	80.88	80.88	83.09
MMT	semantic farthest	**80.59**	**88.82**	**88.68**	**90.59**	**91.18**
	optimal difficulty	72.79	75.00	77.21	79.41	79.41
	random	76.47	76.47	85.29	86.76	88.24
SEED	semantic farthest	**88.97**	**91.62**	**91.76**	**92.21**	**92.35**
	optimal difficulty	85.29	88.23	86.76	90.44	88.97
	random	87.50	84.56	91.91	90.44	90.44

Table 12: **Comparison of different candidate question selection strategies.** "Semantic farthest" means selecting questions with the largest semantic distance, "optimal difficulty" means selecting questions with the smallest difficulty distance, and "random" means purely random selection.

Benchmark	# Candidate	Evaluation Data Budget				
		1%	2%	3%	4%	5%
AI2D	5	86.62	**91.03**	**93.24**	**94.71**	**95.15**
	7	**89.71**	90.44	91.91	92.65	94.12
	10	81.62	88.97	92.65	92.65	93.38
MMMU	5	**77.79**	81.47	83.38	**84.71**	**87.50**
	7	**77.79**	77.94	80.88	84.56	86.03
	10	**77.79**	**81.62**	**83.82**	84.56	85.29
MMT	5	80.59	**88.82**	88.68	90.59	91.18
	7	82.35	88.24	88.24	**91.18**	**91.91**
	10	**83.82**	87.50	**88.97**	88.97	90.44
SEED	5	**88.97**	**91.62**	**91.76**	**92.21**	**92.35**
	7	86.76	87.5	88.97	89.71	89.71
	10	86.76	87.5	88.24	88.24	88.24

Table 13: **The impact of different number of candidate questions.**

Candidate Question Selection Strategy As stated in Equation 2 in Section 2.2, we select questions with the largest semantic distance as the candidates ("semantic farthest"). To demonstrate the superiority of our approach, we compare it against two widely adopted question selection baselines: random sampling ("random"), and selecting questions with the smallest difficulty distance ("optimal difficulty"). As summarized in Table 12, while the "optimal difficulty" strategy achieves the best performance on the AI2D benchmark, its effectiveness does not generalize well across other benchmarks. In contrast, the "semantic farthest" strategy demonstrates consistently strong performance across all evaluated benchmarks and under different compression ratios. Therefore, we choose to use the semantic farthest strategy, as it not only exhibits broad applicability and consistent performance across diverse benchmarks, but can also introduce greater informational diversity.

Method	MMMU	MMT
Baseline (best)	85.3	90.4
1PL	**87.5**	**91.2**
2PL	86.8	**91.2**
3PL	85.3	89.7

Table 14: Performance of A^2-Judger with different IRT model configurations.

Expansion of Candidate Question Pool We investigate the impact of the number of candidate questions (see Equation 2) on the performance of A^2-Judger by expanding $|C_k^*|$ from 5 to 7 and 10. As shown in Table 13, a larger candidate set introduces more flexibility, but also brings additional noise, making it harder to identify the optimal next question.

C.3 The Impact of IRT Models

In A^2-Judger, question difficulty is estimated using a 1PL model, combined with semantic features for item selection. To investigate whether incorporating additional IRT parameters improves performance, we consider more complex models: 2PL, which models item discrimination, and 3PL, which additionally accounts for guessing.

We conducted experiments on two representative benchmarks, and the results are summarized in Table 14. The results indicate that increasing the complexity of the IRT model does not consistently improve recommendation performance (A^2-Judger still outperforms the strongest baseline across all settings). In these multi-modal, high-dimensionall settings, additional parameters may lead to overfitting or provide limited benefit. Therefore, the 1PL model offers a favorable balance between simplicity and effectiveness for A^2-Judger.

C.4 Extension of Data Budgets

To comprehensively evaluate the scalability and robustness of A^2-Judger, we extend the empirical analysis across a data budget spectrum ranging from 1% to 10% on the AI2D, MMMU, and MMT-Bench benchmarks (SEEDBench is omitted due to the high cost considering its large scale). The results, detailed in Table 15, Table 16, and Table 17, demonstrate that A^2-Judger consistently maintains a performance advantage over all baselines across most data budget (e.g. 4%, 5%, and 10%). We observe that the ranking accuracy exhibits monotonic convergence toward the full benchmark performance as the evaluation budget increases. Furthermore, the framework demonstrates substantial efficiency gains under normalized computational expenditures; specifically, A^2-Judger at a 5% budget yields significantly higher accuracy than a 10% random sampling strategy, as evidenced by the 88.7% versus 85.7% accuracy comparison on MMMU. Collectively, these findings justify the strategic selection of a 5% budget for our primary experiments, as it serves as a representative extreme low-sample threshold that strikes an optimal balance between substantial computational reduction and the retention of sufficient information density for reliable model ranking.

D Analysis of A^2-Judger from Additional Dimensions

In this section, we analyze A^2-Judger from additional perspectives, demonstrating that beyond the findings elucidated in the main text, the method possesses other desirable properties.

D.1 Linear Preservation of Relative Accuracy Differences

The inversion counts presented in the main experiments primarily characterize the relative ranking of model capabilities; however, they may not directly substantiate A^2-Judger's ability to accurately quantify the capability gaps between models. To address this, Table 18 presents the Pearson correlation coefficients between the pairwise ability differences predicted by A^2-Judger and the actual accuracy differences observed on the full benchmark. As shown, the quantitative disparities in model capabilities are effectively preserved across various datasets and sampling ratios.

E Case Study

As an agent-driven evaluation framework, A^2-Judger offers a major advantage in enhancing the interpretability of assessment results. We provide two representative examples in Figure 5 and Figure 6. These cases illustrate how information stored in the dynamic memory enables the agent to efficiently analyze the evaluated model's performance across different types of questions (highlighted in blue text in the figures), thereby guiding more informed selection of subsequent evaluation items. Furthermore, the combination of model ability estimation and corresponding question difficulty analy-

Method	Ranking Accuracy (↑)						Ranking Stability (↑)					
	1%	2%	3%	4%	5%	10%	1%	2%	3%	4%	5%	10%
Random	84.7	90.0	89.6	91.3	91.6	94.0	77.6	84.0	86.0	89.0	89.0	92.3
Stratified	83.5	91.2	91.8	93.4	93.5	94.9	75.0	85.6	87.4	89.9	91.7	92.4
Cluster	83.7	89.1	90.9	92.5	93.2	96.0	82.1	86.5	90.1	90.7	91.6	95.9
IRT-Greedy	86.0	89.0	89.7	89.7	89.7	91.9	-	-	-	-	-	-
tinyBenchmarks	81.3	90.7	89.4	90.3	92.9	94.0	72.4	86.5	85.1	84.6	91.2	91.9
Clustering IRT-G	87.6	89.9	90.7	90.7	91.0	93.2	84.3	89.6	92.8	93.1	92.4	94.4
Stratified IRT-G	83.4	87.1	89.3	90.9	93.5	96.0	83.5	90.3	91.2	93.1	95.0	96.3
Hybrid-Greedy	**89.7**	91.2	91.2	93.4	93.4	94.1	-	-	-	-	-	-
A^2-Judger	86.6	91.0	93.2	**94.7**	95.1	**96.8**	95.0	97.5	98.1	98.4	**98.8**	98.5
A^2-Judger$_H$	89.3	**91.8**	**93.4**	94.1	**95.4**	96.6	**97.6**	**98.2**	**99.0**	**98.5**	98.7	**99.6**

Table 15: **Performance of efficient evaluation methods on AI2D across various data budgets (1%–10%).**

Method	Ranking Accuracy (↑)						Ranking Stability (↑)					
	1%	2%	3%	4%	5%	10%	1%	2%	3%	4%	5%	10%
Random	61.9	69.9	77.5	76.6	80.1	85.7	48.7	57.4	69.6	69.0	72.6	81.3
Stratified	50.9	69.6	74.0	72.9	82.6	75.7	39.6	57.6	65.4	65.5	76.2	67.0
Cluster	56.5	62.2	68.8	75.7	77.9	80.4	66.8	61.0	69.6	71.7	73.7	78.1
IRT-Greedy	59.6	74.3	80.1	81.6	81.6	88.2	-	-	-	-	-	-
tinyBenchmarks	58.8	68.4	72.6	75.7	78.7	86.5	45.1	61.5	63.9	67.0	74.4	82.7
Clustering IRT-G	70.9	76.6	80.7	83.8	84.9	89.4	65.1	75.4	78.7	82.1	83.5	89.4
Stratified IRT-G	68.7	75.7	77.8	81.0	83.8	86.9	61.6	68.7	74.3	77.2	82.4	87.5
Hybrid-Greedy	**75.7**	79.4	**83.8**	84.6	85.3	88.2	-	-	-	-	-	-
A^2-Judger	72.8	**81.5**	83.4	84.7	87.5	**91.0**	**100.0**	**97.8**	**97.4**	96.8	96.0	**99.7**
A^2-Judger$_H$	72.8	81.0	83.2	**86.3**	**88.7**	90.9	**100.0**	96.5	97.2	**98.4**	**98.4**	99.3

Table 16: **Performance of efficient evaluation methods on MMMU across various data budgets (1%–10%).**

sis (marked in yellow and orange) assists the agent in identifying the most appropriate questions. Supported by these key components, A^2-Judger can not only evaluate models efficiently, but also provide transparent reasoning behind each evaluation decision. We believe this is an essential step toward building trustworthy and transparent evaluation frameworks for future AI systems.

F Detailed analysis of computational cost

In Section 3.2, our analysis of computational cost focused on both compute cost (GPU seconds per sample) and total monetary cost under the same hardware and software environment. To provide a more intuitive comparison, we conduct our analysis of computational cost focused on theoretical relative overhead, measured by FLOPs. According to the framework we have introduced in Section 2.2, the computational cost of A^2-Judger can be divided into three parts: question difficulty estimation, initialization and iteration.

Question Difficulty Estimation (pre-computed, negligible cost) Before the evaluation begins, A^2-Judger estimates the difficulty of each benchmark question using offline evaluation results from a set of models. These response records are processed with Item Response Theory (IRT) to derive fixed difficulty scores. Since this procedure is performed entirely offline and does not recur during the actual evaluation, its cost is negligible and excluded from the runtime computation overhead.

Initialization (one-time cost) At the start of evaluation, as the ability of the model is unknown, A^2-Judger constructs an initial question pool to build a strong starting point. This involves:

- A^2-Judger computing semantic embeddings for all questions (e.g., via CLIP),
- A^2-Judger performing similarity computation and clustering,
- A^2-Judger sampling a diverse subset of β questions to bootstrap ability estimation,
- the evaluated model solving the selected ques-

Method	Ranking Accuracy (↑)						Ranking Stability (↑)					
	1%	2%	3%	4%	5%	10%	1%	2%	3%	4%	5%	10%
Random	77.1	79.7	81.5	83.7	86.6	90.1	67.1	71.3	74.0	76.8	81.0	85.0
Stratified	64.0	74.6	81.2	83.7	85.3	87.6	51.0	67.2	71.0	76.4	80.1	83.3
Cluster	60.0	70.1	80.0	84.7	87.6	92.6	69.3	77.6	81.3	86.9	87.4	94.6
IRT-Greedy	76.5	80.9	82.4	82.4	88.2	90.4	-	-	-	-	-	-
tinyBenchmarks	64.9	71.6	79.1	82.5	87.5	89.0	52.1	63.7	72.1	73.5	81.8	86.0
Clustering IRT-G	76.2	78.7	84.3	86.2	87.5	89.4	72.5	73.4	80.7	85.0	86.6	89.7
Stratified IRT-G	76.6	80.9	83.5	86.0	87.2	92.6	77.5	81.8	83.8	87.5	89.9	94.1
Hybrid-Greedy	**85.3**	85.3	86.8	86.8	90.4	91.9	-	-	-	-	-	-
A^2-Judger	80.6	**88.8**	**88.7**	**90.6**	**91.2**	**93.8**	**96.3**	**94.7**	95.9	96.2	96.9	98.5
A^2-Judger$_H$	80.6	86.2	87.9	90.4	**91.2**	92.8	95.0	94.3	**96.9**	**97.2**	**98.4**	**99.1**

Table 17: **Performance of efficient evaluation methods on MMT-Bench across various data budgets (1%–10%).**

Threshold	AI2D	MMMU	MMT	SEED
1%	0.89	0.72	0.85	0.93
2%	0.96	0.78	0.89	0.95
3%	0.97	0.83	0.90	0.95
4%	0.97	0.84	0.92	0.96
5%	0.97	0.86	0.93	0.96

Table 18: Transposed performance metrics across different thresholds.

tions (via forwad pass), and

- A^2-Judger generating an intial summary and memory table.

The cost of this one-time procedure is fixed and denoted as F_{init}. Let us take A^2-Judger built upon Qwen2.5-VL-7B to conduct evaluations on the MMT benchmark as an example. The initialization cost involves encoding the CLIP embeddings of questions (6.06 TFLOPs), calculating the pairwise similarity between these questions (15.02 GFLOPs), sampling (negligible), letting the evaluated model solve the β questions (i.e., βF_{model}) and initial summarization (21.52 TFLOPs). Together, these operations amount to about $F_{\text{init}} = \beta F_{\text{model}} + 27.6$ TFLOPs in total.

Iteration (primary computational cost) At each evaluation step, A^2-Judger follows its agent-driven workflow. We denote the computation cost from A^2-Judger as F_{AJ}, which includes:

- *Candidate Retrieval*: filter questions based on current ability estimates and ensure diversity.
- *Question Selection*: the A^2-Judger agent analyzes retrieved candidates, incorporating dynamic memory and IRT-based model ability estimates to pick the most informative next question.
- *Memory Update*: update the memory table to track semantic coverage and difficulty distribution.

Combined with the computation cost of a single *Model Forward Pass* from the evaluated model, denoted as F_{model}, the total per-step iteration computation cost is

$$F_{\text{step}} = F_{\text{model}} + F_{\text{AJ}},$$

Using the MMT benchmark as an example, the computation cost of per-step candidate retrieval (<3 MFLOPs), per-step question selection (19.63 TFLOPs) and per-step memory update (3.62 TFLOPs), totaling $F_{\text{AJ}} = 23.25$ TFLOPs and $F_{\text{step}} = F_{\text{model}} + 23.25$ TFLOPs.

To compare A^2-Judger with full-scale evaluation, we define the relative cost ratio as

$$R(\alpha, \beta, |Q|, F_{\text{model}}) = \frac{(\alpha|Q| - \beta) \cdot F_{\text{step}} + F_{\text{init}}}{|Q| \cdot F_{\text{model}}} = \alpha \cdot \frac{F_{\text{step}}}{F_{\text{model}}} + \frac{F_{\text{init}} - \beta \cdot F_{\text{step}}}{|Q| F_{\text{model}}}$$

where α is the fraction of evaluation questions used, β is the number of questions during initialization and $|Q|$ is the full size of evaluation benchmark. This formula means that, in practice, the relative cost ratio can be conservatively estimated by the fraction α of evaluation questions used and the per-step overhead of A^2-Judger relative to the evaluated model's forward cost. Therefore, the computational overhead introduced by A^2-Judger scales linearly with α and is bounded above by

$$R(\alpha, \beta, |Q|, F_{\text{model}}) \leq \alpha \cdot \frac{F_{\text{step}}}{F_{\text{model}}}.$$

Model	F_{model}	F_{step}	$F_{\text{step}}/F_{\text{model}}$	$R(5\%)$
Qwen2.5-VL-3B	6.04	29.29	4.85×	24.2%
Qwen2.5-VL-7B	13.85	37.10	2.68×	13.4%
Qwen2.5-VL-72B	139.97	163.22	1.17×	5.8%

Table 19: **Computational overhead of A^2-Judger.** F_{model} is the FLOPs of a single forward pass of the evaluated model. F_{step} is the per-step computational cost when evaluated with A^2-Judger. $R(5\%)$ is the relative computational cost under 5% of the data.

This indicates that A^2-Judger achieves significant computational savings compared to full-scale evaluation: by adaptively selecting only a small fraction of questions ($\alpha \ll 1$), the overall evaluation cost can be reduced by an order of magnitude while maintaining reliable ranking consistency.

When we take MMT benchmark as an example, the relative compuation cost is computed as:

$$\begin{aligned} &R(\alpha, \beta, |Q|, F_{\text{model}}) \\ &= \alpha \left(1 + \frac{23.25}{F_{\text{model}}}\right) + \frac{27.6 - \beta * 23.25}{|Q| F_{\text{model}}} \\ &\approx \alpha \left(1 + \frac{23.25}{F_{\text{model}}}\right) \end{aligned}$$

While A^2-Judger substantially reduces the number of evaluation queries by selecting only the most informative ones, it inevitably introduces additional computational overhead. Taking that into consideration, we display the computational cost in Table 19

- **Evaluating a 7B model**: One A^2-Judger iteration incurs about 2.68× the cost of Qwen2.5-VL-7B model forward pass. However, since A^2-Judger achieves high accuracy with only 5% of the full dataset, this translates to a relative cost of 13.4% compared to full-scale evaluation.
- **For larger or smaller models**: As evaluated model size changes, the evaluated model's inference cost adjusts, while the cost of A^2-Judger remains fixed. The relative computational cost on evaluating 3B and 72B models is 24.2% and 5.8% respectively. The advantage is amplified when the evaluated model uses CoT reasoning or when external evaluators (e.g., GPT-4) are invoked for assessment — both of which add significant per-step overhead that A^2-Judger avoids by design.

Overall, A^2-Judger preserves evaluation quality while achieving order-of-magnitude cost savings, making it highly practical for benchmarking in real-world scenarios.

G Prompt of A^2-Judger

Category Identification for Initialization Stage

```
You are an expert educational AI assistant specializing in
      question classification. Your task is to analyze the
      provided questions and categorize them into meaningful
       subject/topic categories.

Task Overview:
You will analyze a set of practice questions (including both
       text and images) and classify each question into a
      meaningful category(Expect two or more question in the
       same category).
The output should be a JSON object mapping question IDs to
      their respective categories.
{
Question ID: # Question ID
Difficulty: # Difficulty
Content: # Content
# IAMAG
Options: # Options
...
}

Output Requirements:
- Return a JSON object with the following format:
{
  "<Question_ID_1>": "<Category_Name_1>",
  "<Question_ID_2>": "<Category_Name_2>",
  ...
}
- Keys are question IDs (index) from the input data.
- Values are descriptive category names that you assign.
- ONLY return the JSON object; do not include any other text
       or explanation.
```

Category Identification for Iteration Stage

```
You are an expert educational classifier. Analyze the
      question and determine its category.
{
Question ID: # Question ID
Difficulty: # Difficulty
Content: # Content
# IAMAG
Options: # Options
...
}

Task: Review the question above. Determine all applicable
      categories from the existing list: {# Category}, or
      include new categories if necessary.

Output Requirements:
- Return a JSON object with:
{"category": ["Existing or new category name(s)"]}
- List ALL relevant categories (minimum 1 item).
- Use EXACT names for existing categories.
- Include multiple entries if needed (e.g., mixed existing/
      new categories).
- Do NOT add explanations, only JSON.
```

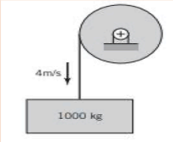
Question Recommendend

```
You are an expert educational AI assistant. Your task is to
      select the most appropriate next question from the
      candidate pool based on:
1. The student's current ability (# ability) estimated by
      IRT.
2. The diversity of question categories in the history.
3. The match between question difficulty and student ability
      .
Prioritize questions that balance category diversity and
      difficulty alignment.

Statistics in history questions
{
# Memory
}
Candidate Question Pool:
{
Question ID: # Question ID
Difficulty: # Difficulty
Content: # Content
# IAMAG
Options: # Options
...
}
Available IDs: # List of Question ID

Output JSON format:
{
    "summary": "Summary the Statistics in history questions
          ...",
    "think": "Reasoning here",
    "question_index": "SELECTED_ID"
}
Only return the JSON object. DO NOT explain.
```

Candidate Question



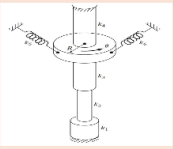
Question (9086): Figure shows a 1000-kg mass being lowered by a cable at a uniform rate of 4 m/s from a ... What additional brake torque is required to bring the system to rest in 0.60 s?<image 1>

Options:

A. 9673J,2218N·m B. 5723J,1218N·m

C. 9673J,1218N·m D. 5723J,2218N·m

Difficulty: -0.423



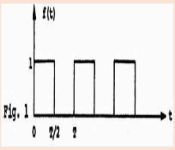
Question (9101): Find the equivalent torsional spring constant of the system shown in <image 1>. Assume that k1, k2, k3, and k4 are torsional and k5 and k6 are linear spring constants.

Options:

A. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_5 + R^2(k_4 + k_6)$ B. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_4 + R^2(k_5 + k_6)$

C. $k_{et} = \frac{k_1k_2k_3}{k_2k_3+k_1k_3+k_1k_2} + k_6 + R^2(k_4 + k_5)$

Difficulty: 0.108



Question (5175): Find the Laplace transform of the periodic waveform shown in <image 1>.

Options:

A. $\frac{1-e^{-s(\frac{T}{2})}}{s(1-e^{-sT})}$ B. $\frac{2-e^{-sT}}{s(1-e^{-sT})}$

C. $\frac{1-e^{-s(\frac{T}{2})}}{s(1-e^{-sT})}$ D. $\frac{1-e^{-s(\frac{T}{3})}}{s(1-e^{-sT})}$

Difficulty: 0.225

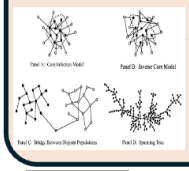
Question (7106): Refer to the figure <image 1>, which term refers to when text and illustration tell different stories, such as in 'Rosie's Walk'?

Options:

A. Contrast B. Reasoning

C. Counterpoint D. Printmaking

Difficulty: -2.181



Question (10700): Epidemiologically speaking, it is crucial to understand the contact structure through ... Which of the following statements best explains Panel C?

Options:

A: One individual's past .. B: It is a structure ...

C: One actor shows ... D: Two populations of ...

Difficulty: -0.402

Ability

Evaluated Model

$\hat{\alpha}_k = -1.09$

A²-Judger

Dynamic Memory

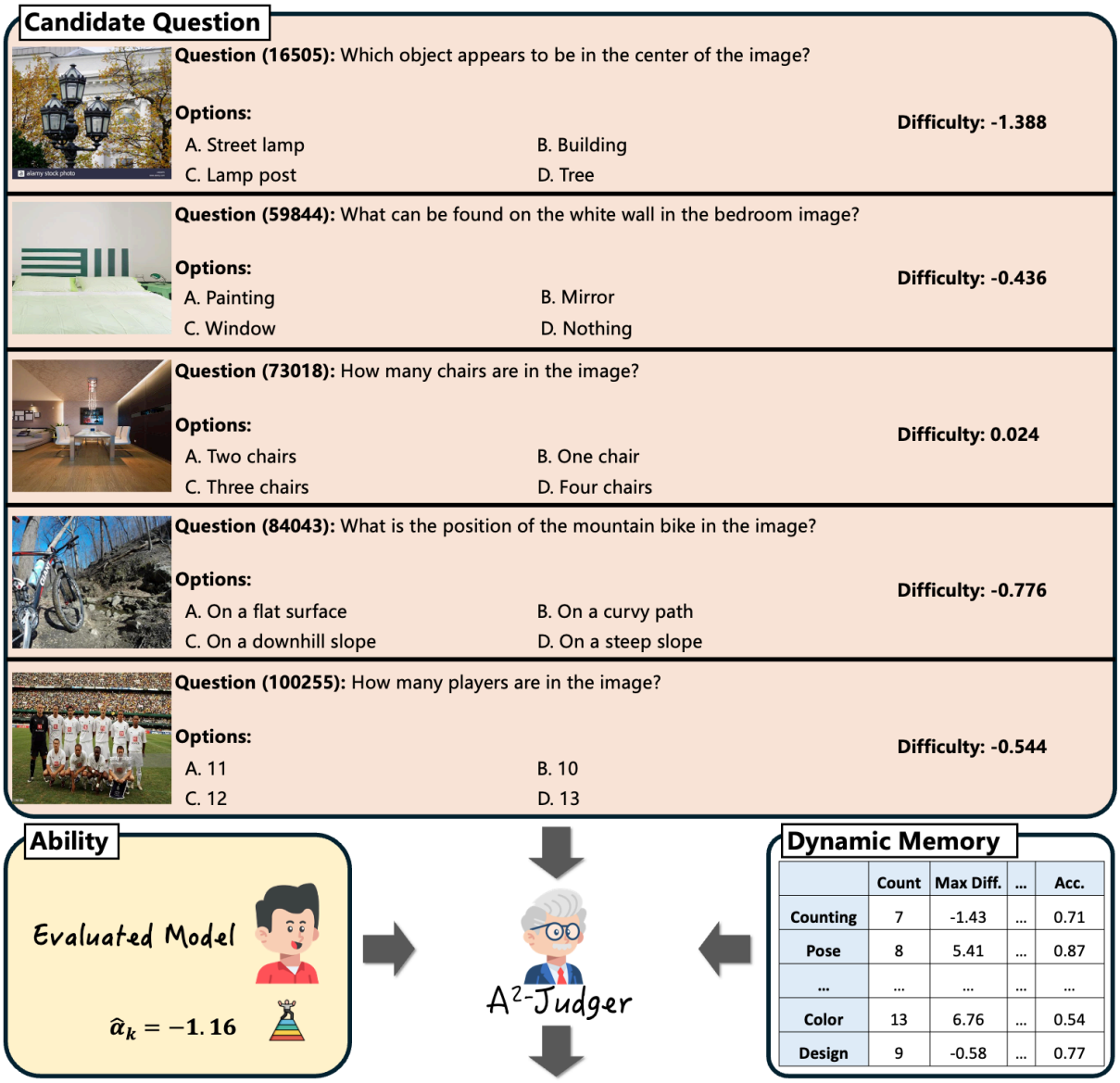
	Count	Max Diff.	...	Acc.
Statistics	15	5.31	...	0.13
Economics	8	-0.61	...	0.25
...	...	...	...	...
History	9	0.61	...	0.68
Physics	17	0.72	...	0.40

"Summary": "The table reflects a comprehensive analysis of question difficulty and accuracy across diverse categories in history and statistics, revealing patterns of variability in student performance. **Categories like 'Art History' and 'Psychology/Social Psychology' show high accuracy rates**, suggesting these topics might be more approachable or familiar to students. Conversely, **subjects like 'Statistics' and 'Biology/Neuroscience' present significant challenges**, as indicated by low accuracy scores and a wide range of difficulty levels. This variability suggests the need for targeted educational strategies that address both foundational understanding and advanced application of concepts, particularly in challenging areas. Additionally, the low engagement with **categories like 'Fashion/Textiles' and 'Clothing/Sleeve Design' indicates potential disinterest or lack of relevance**, which could influence future content development efforts to enhance engagement across all subjects. Such insights highlight the importance of dynamically adapting educational content to meet diverse student needs and interests, as well as enhancing visualization and intuitive concept explanation in complex areas.",

"Think": "Given the **student's ability estimate of -1.09**, it is crucial to select a question that is neither too difficult nor too easy to maintain engagement and foster learning. Considering the question pool, ID 9086 has a **difficulty of -0.42**, closely matching the student's estimated ability. Additionally, this engineering question falls under the 'Engineering/Fluid Mechanics' category, which is relatively balanced in terms of difficulty and performance metrics. This selection strikes a balance between providing an appropriate challenge and ensuring the student encounters a diverse range of categories based on historical performance, **which appears to be lacking in the provided categories**.",

"Question ID": 9086

Figure 5: Response Examples from A^2-Judger on MMMU.



"Summary": "The history of statistics in these questions reveals a diverse range of difficulty levels and content categories. The data shows that certain categories, like **'Image Dominant Color,' are the most frequent but have low accuracy, suggesting they may be challenging or confusing for students**. Other categories, such as **'Sports and Crowd Behavior' and 'Spatial Relations,' offer moderate difficulty levels with moderate accuracy**, implying they might be appropriate for students with varying skill levels. The inclusion of a **'New: Plant Condition Analysis' category indicates exploration in expanding the testing scope**. Balancing difficulty with engagement and accurately assessing a student's ability requires carefully selecting questions from underrepresented categories, such as 'Living Room Layout' or 'Sports and Crowd Behavior.'",
"Think": "Given the **student's estimated ability of -1.16**, we aim to select a question that is neither too easy nor too challenging. The current history shows a distribution across various categories, and we should prioritize balancing category diversity with appropriate difficulty alignment. Questions with difficulties close to -1.16 are ideal. 'Question ID: 16505' has a **difficulty of -1.39**, which is **close to the student's ability level, and its content (determining the central object in an image) is balanced and straightforward**. Other options, such as 'Question ID: 59844' or 'Question ID: 84043,' have slightly higher difficulty levels but could still be within reach.",
"Question ID": 16505

Figure 6: Response Examples from A^2-Judger on SeedBench.