C²RBench: A Chinese Complex Reasoning Benchmark for Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) have achieved remarkable progress in autonomous reasoning, evolving from basic text processing to sophisticated multimodal reasoning, a critical capability for general-purpose AI assistants. However, existing benchmarks usually fail to adequately capture the intricate multi-step reasoning demands inherent in real-world scenarios. To bridge this gap, we propose **C²RBench**: a **C**hinese **C**omplex **R**easoning **Bench**mark for evaluating multi-step, multimodal advanced reasoning capability of LLMs. C²RBench comprises 1,115 carefully curated Chinese tasks, which are organized into eight domain-specific subsets, each meticulously designed to mirror real-world challenges. This hierarchical benchmark features three difficulty tiers based on the number of reasoning steps required (average 8.44 steps per task), significantly exceeding existing benchmarks in cognitive complexity. Extensive evaluations of 16 LLMs (including DeepSeek-R1) and 20 multimodal large language models (MLLMs) on C²RBench reveal critical performance gaps: GPT-4o achieves only 45.20% accuracy, indicating substantial room for improvement.

1 Introduction

Recent advances in reasoning language models like o1-preview[1] and DeepSeek-R1 (DeepSeek-AI et al., 2025) have greatly improved the reasoning capacity of large language models (LLMs), enabling LLMs to engage in more extended and deliberate thought processes to tackle increasingly complex problems compared to earlier LLMs (Yao et al., 2023). These reasoning models exhibit improved multi-step reasoning, decision-making, and adaptability across diverse tasks (Chen et al., 2024).

However, existing reasoning benchmarks usually fail to satisfy the evaluation requirements of reasoning language models (Huang and Chang, 2023).

[1]https://openai.com/o1/

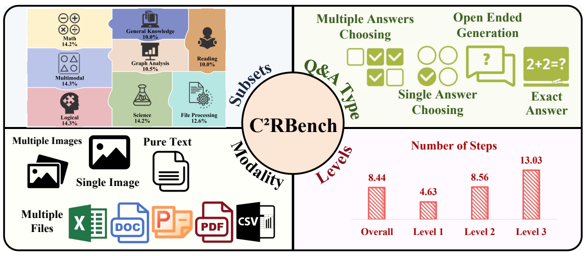


Figure 1: Illustration of the features of C²RBench.

First, these benchmarks often focus on a small number of reasoning tasks with limited reasoning steps. For instance, GSM8K (Cobbe et al., 2021) assesses elementary mathematical operations involving only 2 to 8 reasoning steps.

Second, current benchmarks, such as DRCD (Shao et al., 2019) and C-Eval (Huang et al., 2023), lack alignment with real-world problem-solving scenarios.

Third, many reasoning benchmarks suffer from modality and language constraints as they usually focus on textual reasoning in English, neglecting other modalities and languages (Li et al., 2024b; Huang et al., 2024b).

To mitigate these challenges, we propose the **C**hinese **C**omplex **R**easoning **Bench**mark (**C²RBench**) for evaluating multi-task, multi-step and multimodal reasoning capability of LLMs. As illustrated in Figure 1, it features a broader and more reasoning steps than most existing benchmarks. Specifically, C²RBench favors multi-step reasoning, with an average of 8.44 sequential reasoning steps per task. It is designed with three levels of reasoning complexity in terms of the number of reasoning steps required, where level-3 reasoning tasks consist of 13.03 reasoning steps on average. It is also developed to support multimodal reasoning across text, images, tables and files. Reasoning tasks selected in C²RBench are usually from real-world scenarios, e.g., file processing, graph analysis, reasoning in science, in order to simulate evaluation of LLM reasoning in real-world applications. Despite being related to a previous reasoning benchmark GAIA (Mialon et al., 2023), C^2RBench is larger (1,115 vs 466 reasoning tasks) and more aligned to real-world scenarios than GAIA with a different language focus (Chinese vs English).

The main contributions of the C^2RBench can be summarized as follows:

1. We curate a challenging multimodal multi-step reasoning benchmark in Chinese, addressing the critical gap in Chinese reasoning benchmarks.

2. The curated benchmark C^2RBench benefits from its high cognitive complexity and close alignment with real-world reasoning scenarios. It consists of 1,115 tasks across 8 subsets and 3 complexity levels, with an average of 4.63 reasoning steps per level-1 task, 8.56 per level-2 task and 13.03 per level-3 task.

3. We conduct extensive evaluations of 16 LLMs and 20 multimodal large language models (MLLMs) on C^2RBench. Experiments reveal critical performance gaps: even state-of-the-art models such as GPT-4o achieve only 45.20% accuracy, suggesting the challenging nature of C^2RBench for advanced LLMs.

2 Related Work

As LLMs continue to advance, the tasks they address have grown more complex (Giadikiaroglou et al., 2024). Reasoning, as a fundamental capability essential for solving complex problems, has increasingly become a focal point of research (Wei et al., 2023; Huang and Chang, 2022; DeepSeek-AI et al., 2025; Team, 2025). Based on the nature of the reasoning tasks, existing benchmarks can be categorized into task-specific benchmarks and comprehensive benchmarks. Compared to task-specific benchmarks designed for particular tasks, comprehensive benchmarks attempt to assess the general reasoning capabilities of LLMs.

Task Specific Reasoning Benchmarks. Many studies evaluate the reasoning capabilities of LLMs across specific dimensions. For instance, HellaSwag (Zellers et al., 2019) and SIQA (Sap et al., 2019) measure LLMs' common-sense reasoning abilities, while MathVista (Lu et al., 2024) and GSM8K (Cobbe et al., 2021) assess their mathematical reasoning skills. SQuAD (Rajpurkar et al., 2018) evaluates LLMs' reading comprehension, whereas ARC (Clark et al., 2018) examines their scientific question-answering capabilities. These benchmarks serve as a foundation for evaluating the specific reasoning capabilities of LLMs.

Comprehensive Reasoning Benchmarks. To provide a more comprehensive and systematic evaluation of MLLMs' reasoning capabilities, several comprehensive benchmarks have been proposed, typically encompassing a diverse set of tasks (Li et al., 2024a). For instance, MMMU (Yue et al., 2024a,b) provides a comprehensive evaluation of MLLMs' performance across tasks in the categories of Art & Design, Business, Science, Health & Medicine, Humanities & Social Sciences, and Technology & Engineering, with tasks presented in a multiple-choice format. Similarly, C-Eval (Huang et al., 2023) is a Chinese benchmark for assessing models' advanced knowledge and reasoning abilities. In addition, both MMBench (Liu et al., 2023) and GAOKAO-MM (Zong and Qiu, 2024) assess MLLMs' visual perception and reasoning abilities, also using multiple-choice tasks. GAIA (Mialon et al., 2023) provides a comprehensive evaluation of MLLMs' capabilities in web search, coding, multimodal tasks, and document reading, with tasks presented in an open-ended generation format. However, these benchmarks still exhibit limited task coverage and lack real-world relevance, constraining their applicability to broad use cases. Moreover, research on comprehensive reasoning evaluation benchmarks in the Chinese domain remains an area yet to be fully explored (Huang et al., 2023). Addressing this gap is crucial for developing LLMs that can effectively handle complex reasoning tasks in Chinese, particularly in multimodal settings.

To address these challenges, we propose C^2RBench, a Chinese dataset designed for multi-task, multi-modal and multi-step reasoning scenarios in real-world applications. Detailed comparison of our dataset with previous datasets is presented in Table 1.

3 Dataset Curation

We follow the pipeline illustrated in Figure 2 to curate C^2RBemch.

3.1 Data Sources and Preprocessing

To maintain applicability in real-world scenarios, we derive our primary data sources from four key categories: (1) published materials (e.g., civil ser-

Datasets	MS Anno?	Size	Lang	Domain	Task Type
MathVista	no	5.14K	en	Math	Open&MC
MMMU	a few	12K	en	Real-world	Open&MC
GAIA	yes	0.5K	en	Real-world	Open
GaoKao-MM	no	0.6K	zh	Exams	MC
C-Eval	no	14K	zh	Exams, Web	MC
DRCD	no	30K	zh	Reading	Open
C^2RBench	**yes**	1.1K	**zh**	**Real-world**	**Open&MC**

Table 1: Comparison of C^2RBench with existing reasoning benchmarks. The column "MS Anno?" indicates whether the dataset provides multi-step reasoning annotations. MC denotes Multi-choice QA, while open refers to open-ended QA.

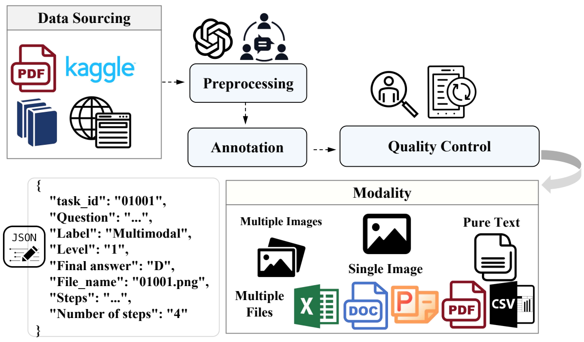


Figure 2: The dataset curation pipeline of C^2RBench.

vice examination question banks, classical literature), (2) authoritative databases (e.g., China National Bureau of Statistics), (3) educational resources (e.g., secondary and post-secondary academic examination questions), and (4) structured datasets from platforms such as Kaggle. Additionally, we employ a hybrid human-AI approach, leveraging ChatGPT for partial data generation (see A.1), followed by rigorous manual screening, verification, and refinement to uphold data quality. This dual strategy—integrating authentic sources with high-quality synthetic data—provides a robust foundation for the real-world applicability of C^2RBench.

Once we collect raw data from multiple sources, we perform preprocessing, which adheres to three key principles. First, we implement strict data construction protocols executed through standardized workflows. Second, all annotators undergo comprehensive training to ensure a thorough understanding of annotation guidelines and consistent labeling practices. Third, we establish a multi-stage quality control strategy featuring periodic progress monitoring and randomized sample audits. Identified issues undergo immediate corrective measures, with error patterns systematically analyzed to prevent recurrence. During preprocessing, we normalize data formats while preserving original semantic structures, ensuring compatibility with diverse reasoning architectures.

3.2 Annotation

Preprocessed data are then transformed into structured formats (please see examples sampled from C^2RBench provided in Appendix 12) adhering to four fundamental question design principles:

- Unambiguous responses: All questions must yield unique, deterministic answers with concise solutions.
- Multi-step reasoning requirements: Each problem necessitates more than 3 interdependent reasoning steps, where intermediate steps demonstrate indispensable logical progression.
- Non-retrievability constraint: Solutions must resist direct retrieval through search engine queries, ensuring genuine reasoning demands.
- Synthetic data validation: ChatGPT-generated content undergo specialized human screening and validation, including semantic consistency checks and logical consistency verification.
- Prevention of Data Contamination: To prevent data contamination, we ensure that no question in C^2RBench can be answered directly through a simple search engine query from.

To operationalize these principles, we implement a dual-layer verification system: automated rule-based filtering complemented by expert review. This framework ensures strict adherence to complexity thresholds while maintaining real-world

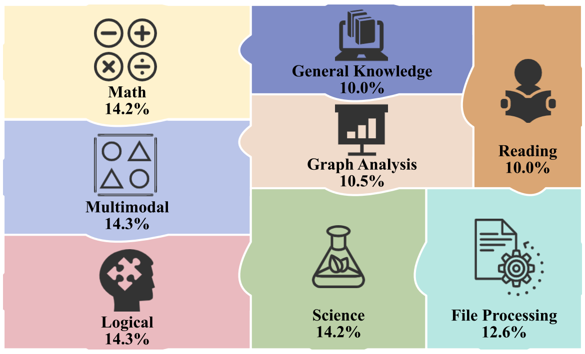


Figure 3: Proportion of different subsets in C^{2}RBench.

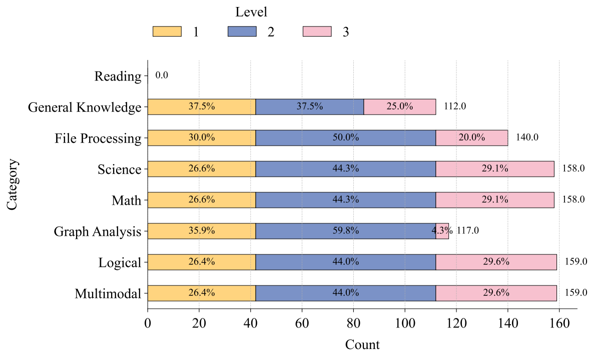


Figure 4: The proportion of different difficulty levels in each subset.

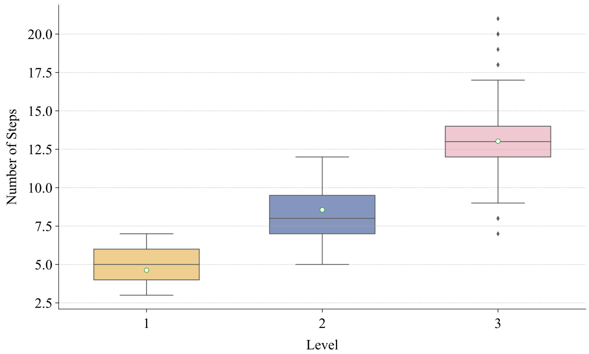


Figure 5: Number of steps in C^{2}RBench.

problem-solving relevance. The final dataset preserves original semantic structures through normalized representation, enabling compatibility with diverse reasoning architectures without compromising task authenticity.

3.3 Quality Control

Following data annotation, we implement a rigorous quality control protocol through cross-verification. This validation process ensures data accuracy and logical consistency through three key metrics: (1) logical coherence of the problem-solving steps, (2) correctness of the final answers, and (3) relevance of supporting materials.

The dataset undergoes systematic iterative refinement via a dual-phase optimization strategy. (1) Deduplication: Qustions with high similarity are removed based on similarity scores (using TF-IDF combined with cosine similarity). Automatically detected duplicates are further verified manually. (2) Dynamic Difficulty Calibration: The complexity of questions is adaptively adjusted based on solver feedback statistics. Our empirical thresholding mechanism automatically elevates problems exceeding 11 procedural steps (originally classified as Level 2) to higher complexity tiers (Level 3), with human validation applied to all reclassifications.

This multilayered quality control framework ensures that the dataset maintains optimal difficulty progression while preserving content diversity and pedagogical validity throughout iterative updates.

3.4 Statistics

After extensive manual annotation and quality control, C^{2}RBench ultimately comprises 1,115 data instances across 8 subsets: Math, Multimodal, Logical, General Knowledge, Graph Analysis, Reading, Science, and File Processing. The proportion of each subset is shown in Figure 3. To enable a more detailed analysis of model reasoning capabilities, we categorize data instances into three difficulty levels based on the number of annotated reasoning steps:

1. **Level 1** (3–6 reasoning steps): Basic reasoning tasks, evaluating fundamental reasoning abilities of LLMs.

2. **Level 2** (7–11 reasoning steps): Intermediate reasoning tasks, assessing the proficiency of LLMs in multi-step reasoning.

3. **Level 3** (12 or more reasoning steps): Advanced reasoning tasks, providing a comprehensive evaluation of the multi-step reasoning and information integration capabilities of LLMs.

We provide detailed statistical data and visual analyses to offer a deep understanding of the structure of C^{2}RBench in Figure 4 and Figure 5.

4 Experiments

We conducted extensive experiments with 16 LLMs and 20 MLLMs on C^{2}RBench to evaluate its effectiveness and utility for assessing reasoning capabilities of LLMs. All models used for evaluation are summarized in Table 2.

Models	Open Source?	Model Size	Multimodal?
DeepSeek-R1 (DeepSeek-AI et al., 2025)	Yes	671B	No
DeepSeek-V2.5-1210 (DeepSeek-AI, 2024)	Yes	236B	No
DeepSeek-V3 (DeepSeek-AI et al., 2024)	Yes	671B	No
Doubao-1.5-vision-pro- [2]	No	undisclosed	Yes
Doubao-pro-32k [2]	No	undisclosed	No
Doubao-vision-pro-32k [2]	No	undisclosed	Yes
Ernie-4.0-Turbo-8k [3]	No	undisclosed	No
Gemini 1.5 Pro [4]	No	undisclosed	Yes
Gemini 1.5 Flash [4]	No	undisclosed	Yes
GLM-4-Long (GLM et al., 2024)	No	undisclosed	Yes
GLM-4v-plus [5]	No	undisclosed	Yes
GLM-Zero-preview [5]	No	undisclosed	No
GPT-3.5-Turbo [6]	No	undisclosed	No
GPT-4o [6]	No	undisclosed	Yes
GPT-4o-mini [6]	No	undisclosed	Yes
GPT-4-Turbo [6]	No	undisclosed	No
hunyuan-turbo-latest [7]	No	undisclosed	No
hunyuan-turbo-vision [7]	No	undisclosed	Yes
Llama-3.2-Vision (Patterson et al., 2022)	Yes	11B, 90B	Yes
LLaVA-OneVision (Patterson et al., 2022)	Yes	7B, 72B	Yes
Moonshot-v1-32k [8]	No	undisclosed	No
Moonshot-v1-32k-vision-preview [8]	No	undisclosed	Yes
o1-mini [6]	No	undisclosed	No
o1-preview [1]	No	undisclosed	No
Pixtral (Agrawal et al., 2024)	Yes	12B	Yes
Qwen2-VL [9]	Yes	2B, 7B, 72B	Yes
Qwen-VL-max (Bai et al., 2023)	No	undisclosed	Yes
QwQ-32B (Yang et al., 2024)	Yes	32B	No
SenseChat-5-1202 [10]	No	undisclosed	No
Spark4.0 Ultra [11]	No	undisclosed	No
Yi-Lightning (Wake et al., 2025)	No	undisclosed	No
Yi-Vision-V2 [12]	No	undisclosed	Yes

Table 2: Models evaluated on the C^{2}RBench.

4.1 Evaluation Settings

We evaluated various models on C^{2}RBench, including both LLMs and MLLMs. For each type, both closed-source and open-source models were considered. All evaluations were conducted in a zero-shot setting to assess the ability to generate accurate answers without fine-tuning or reliance on few-shot examples.

We used accuracy (ACC) as the evaluation metric. Given that the dataset contains a large number of open-ended and multiple-choice questions, simple regularized matching is deemed unsuitable. Therefore, we employed GPT-4o as the evaluation model, and through random sampling and manual inspection, the evaluation accuracy exceeded 98%. The evaluation process is illustrated in Figure 6.

All experiments were conducted on servers containing eight NVIDIA A6000 GPUs with 48GB memory each. When available, we preferentially utilized the official APIs provided by the model developers for experimental evaluations.

To ensure fairness and reproducibility of experimental results, we standardized the setting of `temperature = 0` whenever possible. For models that necessitate a positive sampling temperature, such as Llama, we set `temperature = 0.001` and configured `do_sample = False`. Notably, o1-preview only allows `temperature = 1`, while the temperature setting is ineffective for DeepSeek-R1.

4.2 Tested Models

All tested models are summarized in Table 2.

MLLMs. We evaluated 20 MLLMs on C^{2}RBench. Both closed-source and open-source models were considered. By default, we selected the most recent and highest-performing model for testing for each model family, such as GPT-4o, Qwen-VL and Doubao. Specifically, Llama-3.2-Vision does not officially support Chinese.

Text-only LLMs. We selected 16 highest-performing text-only LLMs, including Deepseek-R1 and o1-preview.

4.3 Results

Overall Performance: The best-performing models, GPT-4o, Doubao-vision-pro-32k and Doubao-1.5-vision-pro-32k, achieve accuracy of 45.20%, 47.62% and 54.98%. On the logical subset, the most powerful reasoning language models, o1-preview and Deepseek-R1, gain an accuracy of 72.33% and 76.1%, highlighting the challenging nature of C^{2}RBench.

Disparity between Open-source and Closed-source Models: The current leading open-source MLLM (e.g., Qwen2-VL-72B-Instruct, as of the paper submission) achieves an accuracy of approximately 37.76%, which is significantly lower than closed-source models like GPT-4o, Doubao-vision-pro-32k, and Doubao-1.5-vision-pro-32k. However, encouragingly, open-source models have shown promising performance on logical reasoning tasks. For instance, Deepseek-R1 achieves an accuracy of 76.10% on the logical subset, surpassing the performance of closed-source state-of-the-art models such as o1-preview.

[2] https://www.volcengine.com/product/doubao
[3] https://ai.baidu.com/ai-doc/WENXINWORKSHOP/am3ih7xdy
[4] https://deepmind.google/technologies/gemini/
[5] https://www.zhipuai.cn
[6] https://openai.com
[7] https://hunyuan.tencent.com
[8] https://platform.moonshot.cn
[9] https://qwenlm.github.io/blog/qwen2-vl/
[10] https://platform.sensenova.cn
[11] https://www.xfyun.cn
[12] https://platform.lingyiwanwu.com

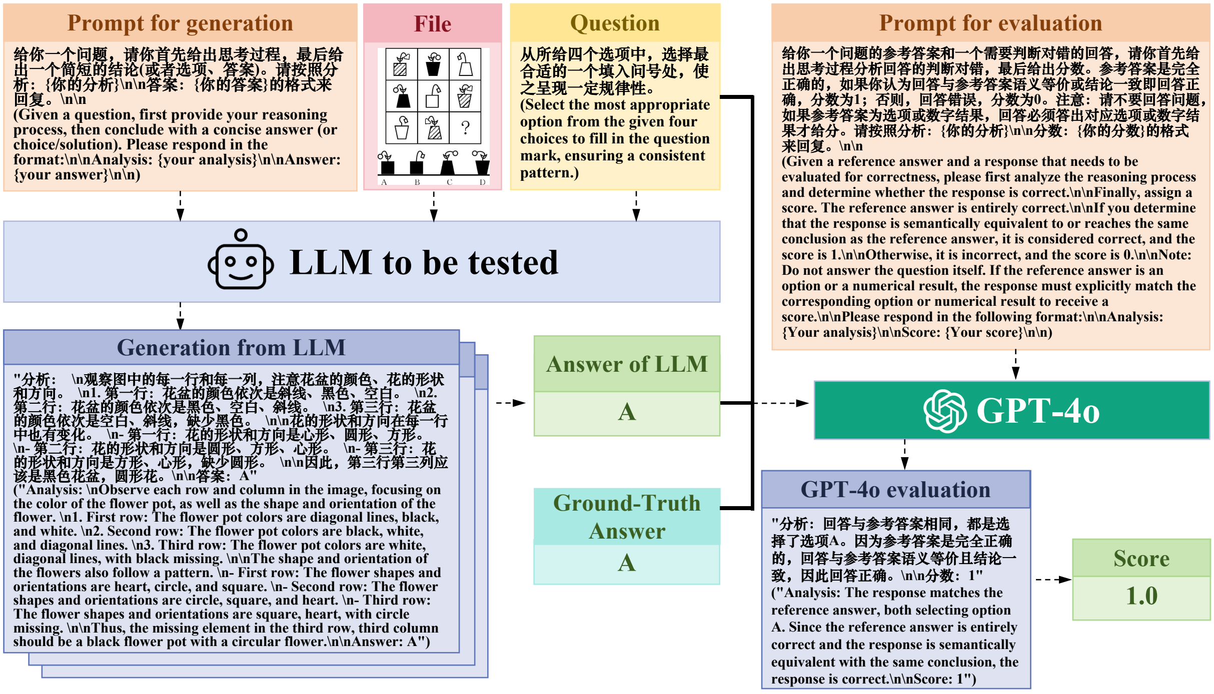


Figure 6: The pipeline for task evaluation in C^2RBench is illustrated using a task example. LLMs represent all the models to be evaluated (see Table 2). The input to the LLM to be tested consists of the prompt for generation, file, and question. The answer of the LLM is derived by extracting its generation. GPT-4o is used as the evaluation model. The input to GPT-4o includes the prompt for evaluation, the answer of the LLM, and the ground truth answer. Finally, the score for the evaluated reasoning task is derived by extracting GPT-4o evaluation results.

Model Performance across Different Subsets: Figure 7 compares the performance of MLLMs across the eight subsets of C^2RBench. Currently, MLLMs still require further attention and improvement in areas such as mathematical reasoning and multimodal inference. Additionally, these models perform relatively poorly on tasks that involve longer reasoning chains, suggesting that future advancements in large models must prioritize enhancing their multi-step reasoning capabilities.

Model Performance across Different Complexity Levels: Figure 8 compares the performance of models across the three levels in C^2RBench. Doubao-1.5-vision-pro-32k demonstrates significantly higher proficiency in Level 1, achieving an accuracy of 64.85%, while open-source models show relatively lower success rates. In Level 2, although the gap narrows, Doubao-1.5-vision-pro-32k still leads with a success rate of 57.55%. In Level 3, the performance differences among models further decrease, indicating that as task complexity increases, the advantages of more advanced models, such as Doubao-1.5-vision-pro-32k and GPT-4o, diminish. This observation highlights current limitations of large language models in handling complex reasoning tasks.

MLLMs Performance on Multimodal Reasoning: Figure 9 presents the evaluation results of MLLMs on the multimodal subset. The results show that open-source models, such as Pixtral-12B, have already surpassed closed-source state-of-the-art models like GPT-4o in multimodal reasoning. Furthermore, the performance gap among models is relatively narrow, while overall accuracy remains low. This underscores the substantial room for improvement in the multimodal, multi-step reasoning capabilities of current MLLMs.

Text-only LLMs Performance on Logical Reasoning: Figure 10 presents the evaluation results of text-only LLMs on the logical reasoning subset. These results primarily target existing large reasoning models, where even the state-of-the-art models, such as o1-preview and DeepSeek-R1, achieve an accuracy of only 72.33% and 76.10%. These results underscore the significant challenges posed by C^2RBench in advanced logical reasoning tasks.

We further investigated DeepSeek-R1's performance through web interface testing, obtaining an accuracy of 74.48% on the logical reasoning subset. Detailed inference time analysis (Table 3) re-

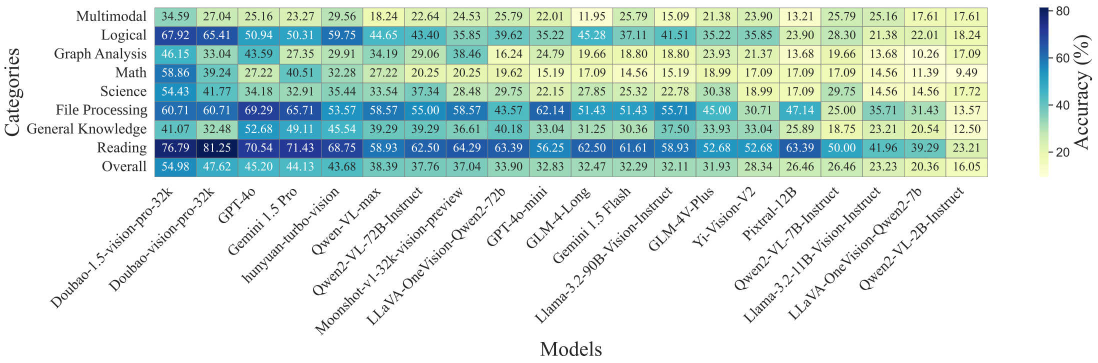


Figure 7: MLLMs' performance across different subsets.

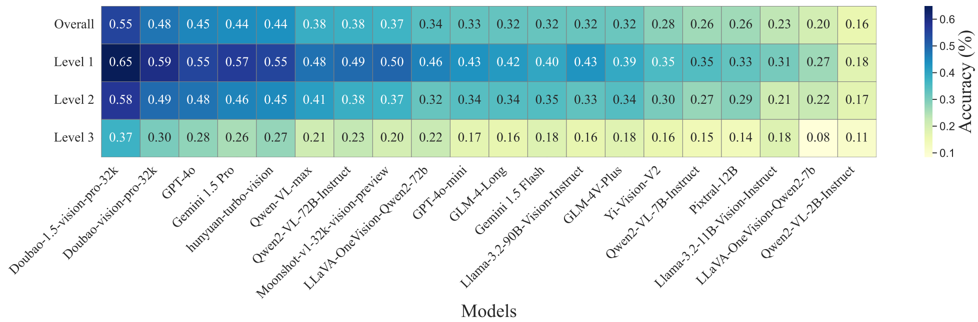


Figure 8: MLLMs' performance across different levels.

veals a positive correlation between reasoning time and problem difficulty level. Notably, incorrect responses consistently require longer processing time (mean = 285.83s) compared to correct solutions (mean = 100.69s), aligning with established findings in reasoning models (Huang et al., 2024a). This observed pattern highlights the critical challenge of persistent reasoning loops in LLMs, suggesting that mitigating unproductive computational cycles represents a key direction for future research.

Times (s)	All	Correct	Wrong
Overall	147.26	100.69	285.83
Level 1	83.57	63.88	167.25
Level 2	132.49	91.98	281.00
Level 3	226.19	158.37	345.88

Table 3: Average reasoning time per task for DeepSeek-R1 on the logical reasoning subset.

5 Analysis

We further conducted an in-depth error analysis and case study to gain deeper insights into the factors underlying the evaluation results for reasoning.

5.1 Error Analysis

To systematically assess the limitations in model performance, we conducted an extensive error typology analysis across four state-of-the-art LLMs.

Error Type	Deepseek-R1	o1-preview	GPT-4o	Qwen2-VL-72B
Perception & Input	0.00%	0.00%	4.42%	3.17%
Knowledge & Comprehension	13.16%	18.92%	52.21%	52.88%
Logic & Reasoning	97.37%	97.73%	69.39%	65.56%
Task Execution	10.53%	0.00%	9.82%	13.40%
Refusal & Inability	0.00%	0.00%	1.96%	7.06%
Generation & Output	0.00%	2.27%	1.31%	2.74%

Table 4: Error type distribution across models. Please note that one output of LLM may involve multiple types of errors (e.g., both Knowledge & Comprehension and Logic & Reasoning), resulting in column totals exceeding 100%. It is important to note that DeepSeek-R1 and o1-preview were evaluated exclusively on the logical subset, while GPT-4o and Qwen2-VL-72B were assessed on the full C^{2}RBench dataset.

The responses from each model were categorized into six distinct error classes through a multi-stage annotation process. (1) **Perception & Input**: Failures in processing multimodal inputs. (2) **Knowledge & Comprehension**: Gaps in domain-specific knowledge. (3) **Logic & Reasoning**: Breakdowns in multi-step reasoning chains. (4) **Task Execution**: Procedural errors in problem-solving. (5) **Refusal & Inability**: Unwarranted non-responses. (6) **Generation & Output**: Errors in formatting or linguistic generation. Note: Individual problems may exhibit multiple error types

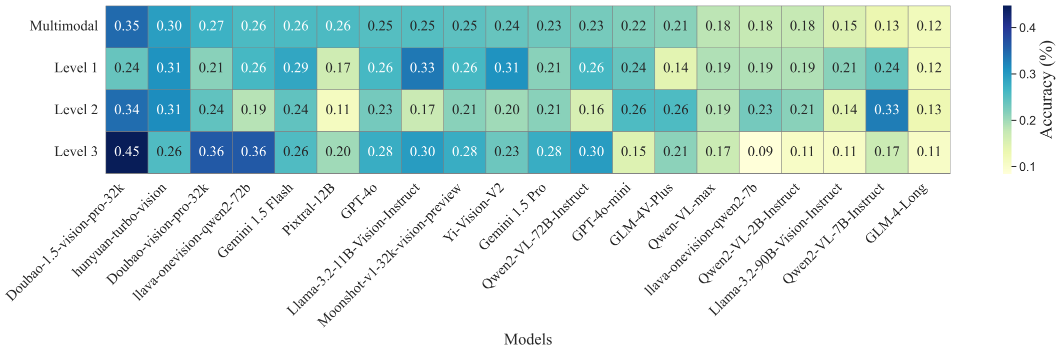


Figure 9: Results of MLLMs on the multimodal reasoning subset.

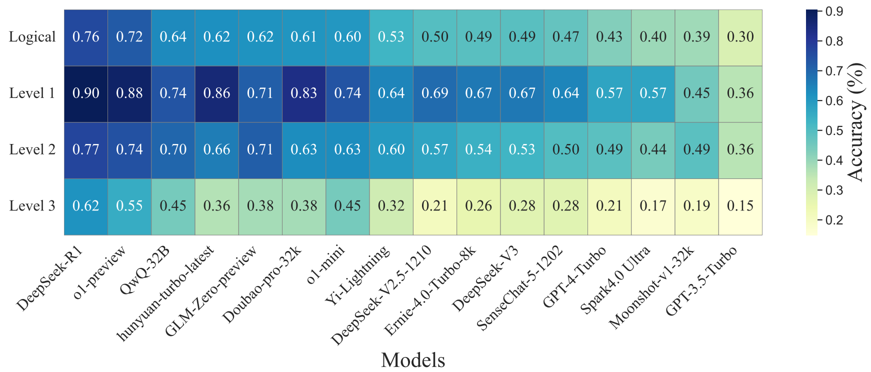


Figure 10: Results of text-only LLMs on the logical reasoning subset.

simultaneously. Full definitions and prompt for error analysis are provided in Appendix A.2.

Table 4 highlights two critical needs for LLMs in Chinese reasoning: improved cross-modal understanding to resolve perceptual-input mismatches, and more efforts dedicated to multi-step reasoning. The persistent prevalence of logical errors (exceeding 65% across all LLMs) particularly underscores the urgency of addressing these challenges.

5.2 Case Study

Figure 13 presents a multimodal case from C^{2}RBench, showcasing responses from four models: two high-performing closed-source models, Doubao-1.5-Vision-Pro-32k and GPT-4o; and two high-performing open-source models, Qwen2-VL-72B and Llama-3.2-90B-Vision. This multimodal reasoning example effectively facilitates a comparative analysis of the performance differences among existing MLLMs.From the responses, it can be observed that the errors made by Qwen2-VL-72B and Llama-3.2-90B-Vision stem from insufficient fine-grained visual analysis capabilities.

Figure 14 illustrates a logical reasoning example from C^{2}RBench, showcasing responses from four models: one high-performing closed-source large reasoning model, o1-preview; one high-performing open-source large reasoning model, DeepSeek-R1; Hunyuan-turbo-latest and GPT-4o. This example highlights the differences in reasoning steps across these models, revealing variations in their multi-step reasoning capabilities.

6 Conclusion

In this paper, we have presented C^{2}RBench, a benchmark designed to evaluate the multi-step reasoning capabilities of LLMs in Chinese. C^{2}RBench consists of 1,115 tasks organized into eight distinct subsets. These tasks are categorized into three difficulty levels based on the number of steps involved. The key highlights of C^{2}RBench are its real-world scenarios, extensive task coverage, and its focus on Chinese. Our core contribution lies in providing a comprehensive and challenging benchmark for assessing LLMs' performance on these complex, multi-step tasks in Chinese. Experimental results of 20 MLLMs and 16 text-only LLMs indicate that C^{2}RBench remains challenging for state-of-the-art models. By systematically assessing complex reasoning across eight subsets, C^{2}RBench establishes a rigorous and comprehensive benchmark.

Limitation

Although C^{2}RBench is comprehensive, like any benchmark, it has its limitations. The manual curation process may introduce biases, and its focus on high-difficulty, multi-step reasoning problems may not fully capture the capabilities required for LLMs. However, we believe strong benchmark performance is essential for evaluating advanced LLMs. The reasoning challenges posed by C^{2}RBench are evident from the performance of 20 MLLMs and 16 text-only LLMs. To balance complexity and practicality, the dataset includes multiple-choice, single-choice, calculation-based tasks, and concise open-ended questions.

References

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khandelwal, Timothée Lacroix, Guillaume Lample, Diego Las Casas, Thibaut Lavril, Teven Le Scao, Andy Lo, William Marshall, Louis Martin, Arthur Mensch, Pavankumar Muddireddy, Valera Nemychnikova, Marie Pellat, Patrick Von Platen, Nikhil Raghuraman, Baptiste Rozière, Alexandre Sablayrolles, Lucile Saulnier, Romain Sauvestre, Wendy Shang, Roman Soletskyi, Lawrence Stewart, Pierre Stock, Joachim Studnia, Sandeep Subramanian, Sagar Vaze, Thomas Wang, and Sophia Yang. 2024. Pixtral 12b. *Preprint*, arXiv:2410.07073.

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. *arXiv preprint arXiv:2308.12966*.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. 2024. M^{3}CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8199–8221, Bangkok, Thailand. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei

Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Panagiotis Giadikiaroglou, Maria Lymperaiou, Giorgos Filandrianos, and Giorgos Stamou. 2024. Puzzle solving using reasoning of large language models: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11574–11591, Miami, Florida, USA. Association for Computational Linguistics.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024a. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*.

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

Bo Li, Peiyuan Zhang, Kaichen Zhang, Fanyi Pu, Xinrun Du, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. 2024a. Lmms-eval: Accelerating the development of large multimoal models.

Jian Li, Weiheng Lu, Hao Fei, Meng Luo, Ming Dai, Min Xia, Yizhang Jin, Zhenye Gan, Ding Qi, Chaoyou Fu, Ying Tai, Wankou Yang, Yabiao Wang, and Chengjie Wang. 2024b. A survey on benchmarks of multimodal large language models. *Preprint*, arXiv:2408.08632.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2023. Mmbench: Is your multi-modal model an all-around player? *arXiv:2307.06281*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *Preprint*, arXiv:2311.12983.

David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2022. The carbon footprint of machine learning training will plateau, then shrink. *Preprint*, arXiv:2204.05149.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *Preprint*, arXiv:1806.03822.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473.

Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. 2019. Drcd: a chinese machine reading comprehension dataset. *Preprint*, arXiv:1806.00920.

Kimi Team. 2025. Kimi k1.5: Scaling reinforcement learning with llms.

Alan Wake, Bei Chen, C. X. Lv, Chao Li, Chengen Huang, Chenglin Cai, Chujie Zheng, Daniel Cooper, Fan Zhou, Feng Hu, Ge Zhang, Guoyin Wang, Heng Ji, Howard Qiu, Jiangcheng Zhu, Jun Tian, Katherine Su, Lihuan Zhang, Liying Li, Ming Song, Mou Li, Peng Liu, Qicheng Hu, Shawn Wang, Shijun Zhou, Shiming Yang, Shiyong Li, Tianhang Zhu, Wen Xie, Wenhao Huang, Xiang He, Xiaobo Chen, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Yanpeng Li, Yongke Zhao, Yongzhen Luo, Yuchi Xu, Yuxuan Sha, Zhaodong Yan, Zhiyuan Liu, Zirui Zhang, and Zonghong Dai. 2025. Yi-lightning technical report. *Preprint*, arXiv:2412.01253.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024a. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. 2024b. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Yi Zong and Xipeng Qiu. 2024. Gaokao-mm: A chinese human-level benchmark for multimodal models evaluation. *Preprint*, arXiv:2402.15745.

A Appendix

A.1 Prompts for Dataset Curation

In Figure 11, we present a complete prompt example required to construct a Graph Analysis task using ChatGPT. The generated content is manually reviewed each time to ensure its relevance and usability.

A.2 Prompt for Error Analysis

The GPT-4o-based error analysis prompt as follows:

```
You will be provided with: 1. A
    reference answer that follows
    standard reasoning steps. 2. A model
    -generated response containing
    errors.
Your task is to analyze the errors,
    attribute them to their root causes,
     and classify them into an
    appropriate error type and subtype
    based on the following categories:
1. **Perception & Input Errors** -
    Perceptual Error: Misinterprets
    visual or structured input (e.g.,
    tables, charts). - Input Parsing
    Error: Mishandles input format,
    structure, or symbols (e.g., JSON,
    XML).
2. **Knowledge & Comprehension Errors**
    - Lack of Knowledge: Lacks necessary
     domain knowledge. - Factual Error:
    Provides incorrect or hallucinated
    facts. - Misinterpretation of
    Question Intent: Misunderstands task
     requirements.
3. **Logical & Reasoning Errors** -
    Logical Reasoning Error: Produces
    inconsistent or invalid reasoning. -
     Inductive Reasoning Error:
    Incorrectly generalizes from
    specific cases. - Deductive
    Reasoning Error: Misapplies general
    rules to specific instances. - Long-
    range Dependency Error: Fails to
    maintain key context across
    reasoning steps. - Hypothesis
    Generation Error: Assumes
    unsupported premises. - Causal
    Reasoning Error: Misidentifies cause
    -effect relationships. - Analytical
    Reasoning Error: Struggles with
    multi-step analysis (e.g.,
    mathematical proofs).
4. **Task Execution Errors** -
    Arithmetic Error: Makes numerical
    miscalculations. - Constraint
    Violation Error: Ignores explicit
    constraints (e.g., word limits,
    formatting). - Planning & Omission
    Error: Skips necessary reasoning
    steps or lacks structural coherence.
5. **Refusal & Inability Errors** -
    Overly Conservative Refusal:
    Unjustified refusal despite
    sufficient information. - Failure to
     Answer: States inability to respond
     when reasoning is possible. - Model
     Breakdown: Produces irrelevant,
    incoherent, or nonsensical output.
6. **Generation & Output Errors** -
    Grammar & Syntax Error: Contains
    grammatical or syntactical mistakes.
     - Disorganized Response: Presents
    information in a confusing manner. -
     Hallucination Error: Generates non-
    existent or fabricated content. -
    Redundancy & Repetition Error:
    Repeats content excessively. -
    Ambiguous Response: Provides vague
    or unclear answers.
**Response Format**:
- Analysis: {Detailed analysis of the
    errors, including their root causes}
- Error Type: {Main error category}
- Error Subtype: {Specific error subtype
    }
```

A.3 Examples

In this subsection, we present C^{2}RBench examples from each subset (see Figure 12). Additionally, for the logical (see Figure 13) and multimodal (see Figure 14) subsets, we show responses sampled from four models, with green text representing correct answers and red text representing incorrect ones.

Prompt for data source construction

根据[Technology and the Internet]这个主题，生成[2]个有逻辑上的关联的csv格式数据。根据主题和数据生成标题，将其放在标记"[title]"之后。数据中包含的信息可以根据需要进行扩展和调整，以适应[柱状图]的类型。图表主题是[Technology and the Internet]。每个数据点的差异不应太大。数据的每一行代表一个类别的不同方面。(Generate two logically related CSV data points based on the theme of [Technology and the Internet]. Create a title based on the theme and the data, and place it after the tag "[title]". The information contained in the data can be expanded or adjusted as needed to fit the type of [bar chart]. The chart's theme is [Technology and the Internet]. The differences between each data point should not be too large. Each row of the data represents different aspects of a particular category.)

Prompt for generating graph image files

假设您是专业的Python图形学家。请使用python根据以下数据绘制并保存图表，图像必须清晰直观。代码不应有额外的缩进。将给定数据转换为三个变量：data_labels、数据line_labels。Data_labels表示除第一列之外的每一列的标签。Line_labels表示除第一行之外的每一行的标签。数据表示数据中的数值数组。使用[柱状图]类型绘制数据。[绘制所选图表类型的基本设置。[绘制代码的具体设置，以增加多样性。图像必须另存为[path_to_save].png。请除代码末尾的当前图像状态。在savefig()之前按tight_layout()自动调整图像大小。图的标题应为[生成的图表标题]。如果图片中的字符串太长，请找到一种让所有字符都显示的方法，而不是被覆盖和堆叠在一起。生成的代码的开头和结尾不要有多余的前导词，例如"python code、python、""等。检查生成的代码以确保它不会报告错误，不包含未定义的函数。数据文件如下：(Assume you are a professional Python graphics expert. Please use Python to plot and save the chart based on the following data. The image must be clear and intuitive. The code should not have unnecessary indentation. Convert the given data into three variables: data_labels, line_labels. Data_labels represents the labels for each column except the first one. Line_labels represents the labels for each row except the first one. The data represents the array of values in the dataset. Plot the data using the [bar chart] type. [Plot the basic settings for the selected chart type.] [Specific settings for the plot code to add variety.] The image must be saved as [path_to_save].png. Clear the current image state at the end of the code. Automatically adjust the image size with tight_layout() before calling savefig(). The chart title should be [Generated Chart Title]. If the strings in the image are too long, find a way to display all characters without them being covered or stacked together. The generated code should not contain unnecessary leading words at the beginning and end, such as "python code", "python", etc. Check the generated code to ensure it will not report errors or contain undefined functions. The data file is as follows:)

Prompt for generating QA

根据输入的具有给定标题和图表类型的csv数据，设计一个从两个图表中的信息才能够得到答案的hard难度的问答对，可以直接通过数据回答。每个答案不应包含任何提示、解释或注释等。答案应当是唯一的。并写出推理得到答案的过程链。(Design a hard-level question-answer pair based on the input CSV data with a given title and chart type, where the answer can only be obtained by combining information from both charts. The answer should be directly derived from the data. Each answer should not contain any hints, explanations, or comments. The answer must be unique. Also, provide the reasoning process chain that leads to the answer.)

Figure 11: Example prompts for constructing a Graph Analysis task using ChatGPT.

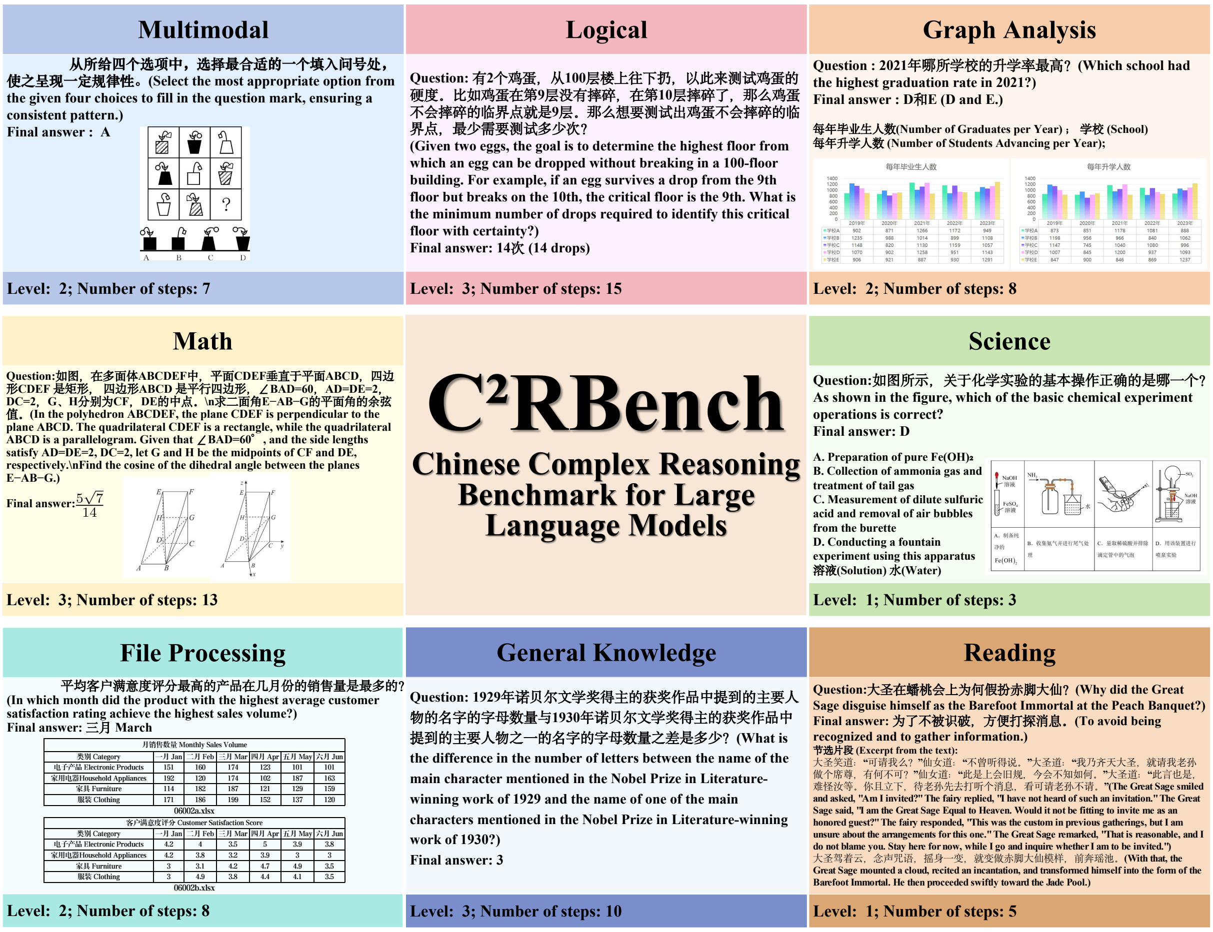
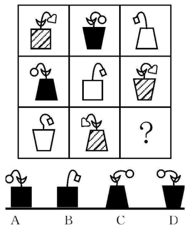


Figure 12: Sampled C^{2}RBench examples from each subset.

"Question":"从所给四个选项中，选择最合适的一个填入问号处，使之呈现一定规律性。",
("Select the most appropriate option from the given four choices to fill in the question mark, ensuring a consistent pattern.")
"Label": "Multimodal", "Level": "2", "Final answer": "A",
"Steps": "1.观察题目，本题考察图形轮换规律。\n2.观察每行的各个图形的元素（包括形状、花纹、黑白）在下一行中都重复出现。\n3.分析"？"处的图形特征：\n4.花盆应为黑色正方形，排除C、D项；\n5.花茎上应带有月牙形状；\n6.花朵应为圆形。\n7.因此，本题正确答案为A选项。",
("1.Observe the problem. This question assesses the pattern of graphical rotation.\n2.Observe that the elements of each shape in a row (including shape, pattern, and color) reappear in the next row.\n3.Analyze the characteristics of the shape at the "?" position:\n4.The flowerpot should be a black square, eliminating options C and D;\n5.The stem should feature a crescent shape; \n6.The flower should be circular.\n7.Therefore, the correct answer is option A.")



Doubao-1.5-vision-pro-32k
1. 首先观察图形特征：... 遍历的规律是每行要出现相同的元素。\n2. 然后具体分析每行元素：...\n3. 最后看选项：.... \n\n分析：通过对每行图形的花盆形状、植物方向和植物顶端形状进行遍历分析，发现第三行缺少黑色梯形花盆、向左方向的植物和圆形顶端的植物，A选项符合这些特征。\n答案：A
(1. First, observe the graphical features: ... The pattern of traversal requires that the same elements appear in each row.\n2.Next, analyze the elements in each row: ...\n3. Finally, examine the options:... \n\nAnalysis: Through a thorough analysis of the pot shapes, plant orientations, and plant top shapes in each row, it is evident that the third row is missing a black trapezoid pot, a left-oriented plant, and a plant with a round top. Option A satisfies these missing elements.\nAnswer: A)

GPT-4o
分析： \n观察图中的每一行和每一列，注意花盆的颜色、花的形状和方向。... \n\n因此，第三行第三列应该是黑色花盆，圆形花。\n\n答案：A

(Analysis: \nObserve each row and column in the image, focusing on the color of the flower pot, as well as the shape and orientation of the flower. ... \n\nThus, the missing element in the third row, third column should be a black flower pot with a circular flower.\n\nAnswer: A)

Qwen2-VL-72B

分析：这是一道图形推理题。...第一行的图形依次顺时针旋转90度，并且颜色交替变化。...根据这个规律，第三行的最后一个图形应该是黑色的，并且是顺时针旋转90度后的形状。\n\n答案：A

(Analysis: This is a figure reasoning question. ... The shapes in the first row rotate 90 degrees clockwise sequentially, with alternating colors. ... Therefore, the third row should also adhere to this rule. Based on this pattern, the last shape in the third row should be black and rotated 90 degrees clockwise.\n\nAnswer: A)

Llama-3.2-90B-Vision
分析：... 图形的颜色有黑色和白色两种。图形的形状有两种：一种是正方形在上，三角形在下；另一种是三角形在上，正方形在下。...根据这些规律，可以推断出问号处的图形应该是黑色三角形在上，白色正方形在下。\n\n答案：D
(Analysis: ... The shapes come in two colors: black and white. There are two possible arrangements of the shapes: one with the square on top and the triangle on the bottom, and the other with the triangle on top and the square on the bottom. ... Based on these patterns, it can be deduced that the shape at the question mark should be a black triangle on top with a white square at the bottom.\n\nAnswer: D)

Figure 13: An example from the **Multimodal** subset. Correct analyses are highlighted in green, while errors are marked in red.

"Question": "在一天的24小时之中，时钟的时针、分针和秒针完全重合在一起有几次，分别是什么时间？",
("In a 24-hour day, how many times do the hour, minute, and second hands of a clock coincide completely, and at what times?")
"Label": "Logical", "Level": "3", "Final answer": "Twice, at 12:00 and 00:00.", "File_name": "",
"Steps": "1. 问题陈述... \n2. 确定角速度... \n3. 分针与时针重合的计算... \n4. 时间换算... \n5. 秒针重合验证... \n6. 三针重合条件：\n 三针重合的唯一时间是在正12点和0点。\n7. 数学证明：... \n8. 计算重合时间：... \n9. 最小公倍数：... 最小公倍数为 11 * 719。\n 重合时间间隔为 11 * 719 * \\(\\Delta\\)(即12小时)。\n10. 首次重合：\n 三针在 t = 0 时重合。\n11. 下一次重合：\n 下一次重合时间：\\(t = 12 \\)小时。\n12. 结论：\n 在24小时内，时针、分针和秒针完全重合的时间只有两次：12点和0点。"
("1. Problem statement... \n2. Determine angular velocities ... \n3. Compute the coincidence of the minute and hour hands ... \n4. Convert time units ... \n5. Verify second-hand alignment ... \n 6. Condition for complete overlap:\n - The only times when all three hands coincide are exactly at 12:00 and 00:00.\n7. Mathematical proof: ... \n8. Compute the coincidence time: ... \n9. Least common multiple: ... The coincidence interval is \\(11 \\times 719 \\times \\Delta \\) (i.e., 12 hours).。First coincidence:\n - The three hands coincide at \\(t = 0 \\).\n11. Next coincidence:\n - The next coincidence occurs at \\(t = 12 \\) hours.\n12. Conclusion:\n - In a 24-hour period, the hour, minute, and second hands coincide completely only twice: at 12:00 and 00:00.", ")

o1-preview
分析：\n时钟的时针、分针和秒针完全重合的时刻，是当它们同时指向12的位置。... 尽管它们的速度不同，但每12小时都会同时回到12的位置。因此，在24小时内，它们会在凌晨12:00:00和中午12:00:00各重合一次。\n\n答案：\n一共2次，分别在凌晨12:00:00和中午12:00:00时刻。
(Analysis: \nThe moment when the hour, minute, and second hands of a clock completely coincide occurs when all three hands point to the 12 o'clock position simultaneously. ... Despite their differing speeds, all three hands align at the 12 o'clock position every 12 hours. Therefore, within a 24-hour period, they coincide twice: once at 12:00:00 AM and again at 12:00:00 PM.\n\nAnswer: \nA total of two times, specifically at 12:00:00 AM and 12:00:00 PM.)

DeepSeek-R1
分析：在24小时内，时针和分针每12小时重合11次，但加入秒针后，只有当秒针也同时指向同一位置时才会三针重合。由于分针和时针的重合时间点并非整秒数（如约1小时5分27秒），此时秒针位置无法与分针、时针同步。只有12点整和0点整时，三个指针完全重合。24小时内，0点和12点各出现一次，共两次。\n\n答案：两次，分别是0时0分0秒和12时0分0秒。
(Analysis: Within 24 hours, the hour hand and minute hand overlap 11 times every 12 hours. However, with the addition of the second hand, the three hands only coincide when the second hand also points to the same position. Since the overlap time of the minute hand and hour hand is not an exact second (for example, around 1 hour, 5 minutes, and 27 seconds), the second hand cannot synchronize with the positions of the minute and hour hands at this moment. The three hands only coincide exactly at 12:00:00 and 0:00:00. Therefore, within 24 hours, this happens twice—once at 0:00:00 and once at 12:00:00.\n\nAnswer: Twice, at 0:00:00 and 12:00:00.)

Hunyuan-turbo-latest
分析：... 要使时针、分针和秒针完全重合，需要在分针和时针重合的时刻，秒针也恰好在 12 点位置。... 一天 24 小时中，时针、分针和秒针完全重合在一起只有两次，分别是 0 点 0 分 0 秒和 12 点 0 分 0 秒。\n\n答案：两次，分别是 0 点 0 分 0 秒和 12 点 0 分 0 秒。
(Analysis: ... For all three hands to coincide, the second hand must also be at the 12 o'clock position at the exact moment when the minute and hour hands overlap. ... it is determined that within a 24-hour period, the hour, minute, and second hands align perfectly only twice: at 12:00:00 AM and 12:00:00 PM.\n\nAnswer: Twice, at 12:00:00 AM and 12:00:00 PM.)

GPT-4o
分析：...因为时针每12小时转一圈，分针每小时转一圈，秒针每分钟转一圈，所以它们在每个整点时刻重合。一天有24小时，因此它们会重合24次。\n\n答案：24次，分别是每个整点时刻(例如：0:00:00, 1:00:00, 2:00:00, ..., 23:00:00)
(Since the hour hand completes one full revolution every 12 hours, the minute hand every hour, and the second hand every minute, they align precisely at the beginning of each hour. Given that there are 24 hours in a day, this alignment occurs 24 times.\n\nAnswer: 24 times, occurring at the start of each hour (e.g., 0:00:00, 1:00:00, 2:00:00, ..., 23:00:00).)

Figure 14: An example from the **Logical** subset. Correct analyses are highlighted in green, while errors are marked in red.