CATATQA: A BENCHMARK FOR TOOL-AUGMENTED LLM QUESTION ANSWERING OVER HETEROGENEOUS CATALYSIS TABLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite their success in general question answering, large language models (LLMs) struggle with hallucinations and inaccurate reasoning in scientific domains. A major challenge stems from experimental data, which are often stored in external sources like supplementary materials and domain-specific databases. These tables are large, heterogeneous, and semantically complex, making them difficult for LLMs to interpret. While external tools show promise, current benchmarks fail to assess LLMs' ability to navigate this data-particularly in locating relevant tables, retrieving key columns, interpreting experimental conditions, and invoking tools. To address this gap, we introduce CataTQA, a new benchmark for **Cata**lytic materials **T**able **Q**uestion and **A**nswering. CataTQA features an automated dataset framework and four auxiliary tools. We evaluate tool-enhanced LLMs across five dimensions: table location, column retrieval, condition analysis, tool calling, and question answering, identifying their strengths and weaknesses. Our work sets a new benchmark for evaluating LLMs in scientific fields and paves the way for future advancements. All data and code are publicly available on GitHub[1].

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable success in a diverse range of fields Bogin et al. (2024); Boiko et al. (2023). Prominently, they have made significant inroads into the medical field Waisberg et al. (2024); Lamb et al. (2024), the materials field Zhang et al. (2024b); Kristiadi et al. (2024), and the industrial field Yang et al. (2023); Saka et al. (2024), revolutionizing traditional practices and driving innovation. Numerous methods, including domain data fine-tuning and chain of thought, enhance LLM question-answering capabilities. However, scientific Q&A applications face specific challenges: (1) Hallucination issue Sadat et al. (2023); Weidinger et al. (2021); Ji et al. (2023); Bubeck et al. (2023): When processing experimental data stored in external sources, LLMs are highly prone to the hallucination issue, which stems from the massive scale, heterogeneous structure, and semantic complexity of data tables. This not only misleads users in decision-making but also undermines the integrity of content in rigorous scenarios like scientific research. (2) Computational limitations Gao et al. (2023); Hendrycks et al. (2021); Lewkowycz et al. (2022); Madaan & Yazdanbakhsh (2022); Nogueira et al. (2021); Qian et al. (2022): Confronted with large-scale heterogeneous tabular data, the computational architectures of LLMs reveal significant limitation, a critical skill required in numerous real-world scientific applications. (3) Inability to identify key decision-making information Ziegler et al. (2025): When confronted with multiple pieces of decision-relevant data, models frequently fail to quickly pinpoint the key information that could enhance the quality and accuracy of their responses.

Regarding the above issues, some research uses LLMs as scheduling models and enhances their Q&A capabilities by leveraging external tools, such as Retrieval-Augmented Generation (RAG) dos Santos Junior et al. (2024); Izacard et al. (2022), math tools Wu et al. (2023); Schick et al. (2023); Lu et al. (2023); Yao et al. (2023), and code interpreters Chidambaram et al. (2024); Wang et al. (2022); Gao et al. (2023). Although LLMs have shown a decent performance in general domain Q&A, their

[1] https://github.com/kg4sci/CataTQA

application in scientific domains reveals significant limitations. Most of the experimental data in the scientific domain exists in the form of structured tables, which are characterized by large data scale, high knowledge intensity, and highly complex content. Existing tool-used evaluation methods Zhuang et al. (2023) are still unable to accurately assess the Q&A performance of large models in the scientific field, which is specifically reflected in two aspects: firstly, there is a lack of effective evaluation of the ability of LLM to quickly locate key tables and fields in massive tabular data; Secondly, a systematic evaluation of the ability of LLM to accurately call external tools for Q&A has not been established.

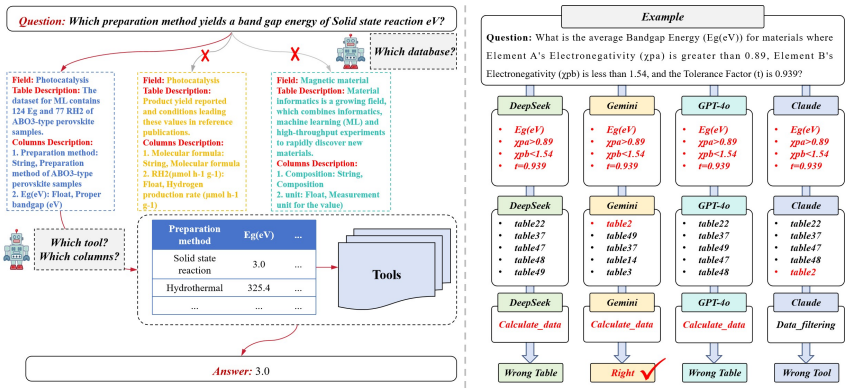

Figure 1: CataTQA has evaluated multiple capabilities of large language models, including the ability to locate key information, the ability to invoke tools, and the ability to answer questions.

To bridge this gap, we propose a Q&A benchmark named CataTQA for fairly evaluating the knowledge localization ability and tool invocation ability of LLM. CataTQA includes 68 catalytic datasets from fields like photocatalysis, with expert-annotated metadata and four defined tool types. Each instance in CataTQA consists of a question, an answer, a series of table requirements and tool for answering questions. As shown in Fig. 1, different from other benchmarks, CataTQA requires LLMs to locate the supporting dataset based on the question and invoke the correct tool for answering.This minimizes the possibility that LLM answers questions merely by recalling its internal knowledge and comprehensively evaluates its key knowledge localization ability and tool usage ability.

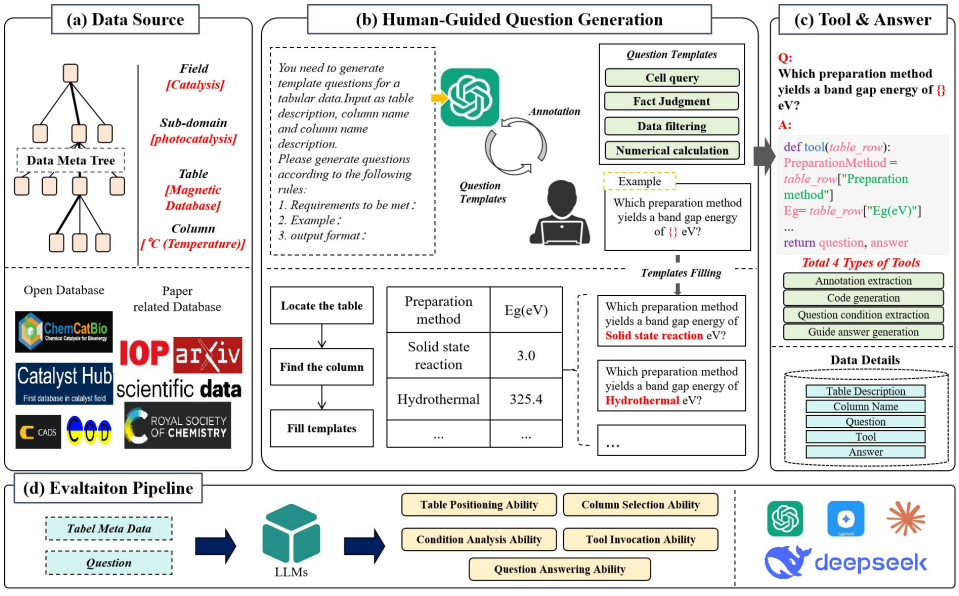

Figure 2: The Construction Process and Application Method of CataTQA, including: (a) Data Source, (b) Human-Guided Question Generation, (c) Tool & Answer and (d) Evaltaiton Pipeline.

The construction and application of CataTQA, as illustrated in Fig. 2, integrate expert annotation and follow a three-phase automated process: **(1) Reference Data Collection:** Based on catalytic field data systems, open-source datasets and paper-associated data are collected. Descriptive metadata is auto-gathered, raw data is preprocessed, and domain experts review/annotate the processed data. **(2) Human-Guided LLM Question Generation:** A template-based approach is used, involving human-guided template creation, validation, and question instantiation with tool attributes. Data table metadata is recorded during question generation to facilitate subsequent LLM localization ability verification. **(3) Tool-Driven Answer Generation:** Answers are derived from datasets via tool operations, ensuring external knowledge is needed. Domain expert annotations provide detailed data table descriptions to validate LLM key information localization capabilities.

Based on this benchmark, we conducted experiments using four proprietary LLMs, each with two prompting ways: Direct Prompting and Code Generation. Our findings indicate that current LLMs have limited proficiency in Q&A tasks based on multi-source tabular data, with significant performance variations across sub-tasks and question types. We decomposed the Q&A process into five key phases: Table Positioning, Column Selection, Condition Analysis, Tool Invocation, and Q&A Ability. While LLMs perform well on some sub-tasks, low accuracy in phases like tool invocation significantly reduces overall Q&A accuracy. The LLMs also show significant variability in handling different question types. For example, they struggle with Fact Judgment questions due to frequent misclassification of task types and poor tool selection.

Our research findings and analysis indicate that CataTQA is a challenging benchmark for existing tool-augmented large language model methods, especially when confronted with difficult questions in the scientific field that require quickly locating effective structured information and conducting more complex reasoning.

2 Related Work

2.1 Benchmarks on Tool-Augmented LLM

Recent advancements in augmenting LLMs with external tools and agents have spurred the development of benchmarks to evaluate their collaborative reasoning and tool-usage capabilities. Existing benchmarks primarily focus on either tool-augmented or table-based Q&A tasks. For instance, ToolBench Xu et al. evaluates LLMs' ability to invoke Application Programming Interfaces (APIs) for task completion, while TAT-QA Zhu et al. (2021) and FinQA Chen et al. (2021) test numerical reasoning over structured tables. ToolQA Zhuang et al. (2023) evaluated LLMs' ability to use external tools based on non overlapping pretraining data, and pointed out possible misuse.However, these benchmarks often operate in isolation, lacking scenarios where LLMs must dynamically coordinate multiple external agents to synthesize information across heterogeneous catalytic tables.

Efforts like WebArena Zhou et al. (2023) and ALFWorld Shridhar et al. (2020) simulate agent-environment interactions but are limited to specific domains (e.g., web navigation) and do not address cross-table reasoning. Meanwhile, benchmarks such as MMTB Koblitz et al. (2021) introduce multi-modal table understanding but omit the integration of external tools. EHRQA Lee et al. (2022) is a practical benchmark on structured EHR data and take a step further towards bridging the gap between text-to-SQL research and its real-life deployment in healthcare.But,a critical gap remains in evaluating LLMs' capacity to orchestrate diverse agents for tasks requiring joint analysis of interconnected tables, domain-specific knowledge retrieval, and multi-step operational workflows.

2.2 Benchmarks on LLMs in Specific Science Field

As the integration of artificial intelligence into different domains accelerates, LLMs are emerging as a dominant force in a growing number of applications Lu et al. (2024); Polak & Morgan (2024); Romera-Paredes et al. (2024); Birhane et al. (2023). Therefore, many benchmarks have emerged for evaluating the performance of LLMs in specified scientific fields Zhang et al. (2024a); O'Leary (2023); Singhal et al. (2023). For instance, MaScQA Zaki et al. (2024) as been classified according to the structure of the questions and subfields of materials science, with the aim of evaluating the understanding ability of LLMs regarding the key concepts of materials science. NASA-QA Bhattacharjee et al. (2024) is an extractive question answering task focused on the Earth science domain. UniDE Ye et al. (2025) represents an innovative framework. It consistently deals with multi-level

dialogue evaluation tasks through the utilization of data produced by LLM for training a small-scale multitask evaluator.

3 CATATQA

3.1 EXPERT ANNOTATIONS

Before evaluating LLMs' ability to answer questions by calling external tools via CataTQA, ensuring their accurate key information localization is essential, as this capability is critical for subsequent complex tasks like data retrieval, inference, and computation. Thus, the benchmark test incorporates descriptive information from catalytic domain datasets (including source, structure, field definitions, and research background) as the foundation for LLMs' localization databases, enabling them to understand and utilize the data effectively. Annotation of tables and fields in datasets is a critical component. In scientific domains (e.g., materials science), specialized data and opaque column nomenclature pose significant comprehension challenges for non-experts, making tabular data and column descriptor annotation paramount (Appendix A.2). Basic metadata annotation enhances LLMs' problem understanding and their ability to identify relevant datasets containing potential answers. Notably, after metadata tagging, dataset expansion has minimal impact on QA efficiency-an approach contrasting with systems like ChatMOF Kang & Kim (2024), which use a table searcher to query all available data, significantly degrading QA efficiency.

During the collection of open datasets, we maintained a policy of minimal data intervention. This approach differs significantly from traditional tabular question answering processing methodologies. Our primary objective was to preserve the authenticity of the original data, a crucial factor for ensuring the reproducibility of table-based analyses. The only data processing performed was the removal of entirely empty rows and columns, a procedure that further demonstrates our commitment to data integrity.For dataset annotation, we engaged domain experts from the National Nanotechnology Center's catalysis field, specifically including an associate researcher and a doctoral student. The annotation process covered three key aspects: 1) the research field classification of each dataset, 2) dataset description, and 3) descriptions of each data field.

3.2 DATASET DETAILS

CataTQA is mainly used to evaluate the comprehensive capabilities of LLMs in the vertical scientific field, specifically in the field of catalysis. These capabilities include the ability to retrieve key data, the ability to invoke tools, and the ability to answer questions. To this end, we have collected and processed data to generate a dataset that meets the following criteria: 1) Ideally, the database should not overlap with the pre-training data of the LLM(The experimental results are shown in the appendix.E.1); 2) It should include accurate metadata information about the dataset, such as the dataset name, dataset description, field names and their descriptions, etc.; 3) The LLM should be able to obtain all the necessary information from the database to answer questions correctly.

CataTQA's data comes from catalysis. To precisely evaluate table question-answering, each instance includes six components: a question, answer, table, relevant columns, conditions, and tools used. The dataset also contains table metadata to assess LLMs' data location ability. It features four question types with two difficulty levels based on the number of answer conditions.

The number of each type of problem in the CataTQA dataset and the average length of each type are shown in the table 1. The four question categories are: 1) **Cell Query,** which addresses the task of locating individual table cells, representing a fundamental table question answering scenario; 2) **Fact Judgment,** judging the facts of the table, about 50% of the error conditions are generated.; 3) **Data Filtering,** which require retrieving results that satisfy multiple specified conditions from the dataset, and 4) **Numerical Calculation,** which involves questions demanding mathematical operations (e.g., median computation, summation, or maximum value identification) to derive correct answers.

Our study categorizes questions as simple or complex. Simple ones have 1-2 constraints and require single-step reasoning. Complex questions contain $\geq$3 constraints, demanding multi-step LLM rea-

soning, which significantly increases difficulty.The metadata[1] and Q&A datasets[2] developed in this study are publicly available through the Hugging Face platform.

Table 1: CataTQA dataset information.

Question Type	Level	Average Length of Questions	Count
Cell Query	simple	12.6	4425
	complex	20.6	4339
Fact Judgment	simple	15.3	5038
	complex	21.4	5756
Data Filtering	simple	14.9	3462
	complex	19.4	1223
Numerical Calculation	simple	13.7	3325
	complex	20.6	2041

Table 2: Comparison of dataset dimensions.

Dataset	#. Tables	Size (GB)	#. Rows	#. Columns	#. Questions
ToolQA Zhuang et al. (2023)	6	5.29	4,356,045	112	1,530
ERATTA Roychowdhury et al. (2024)	7	∼5	–	–	100
DataBench Grijalba et al. (2024)	65	–	3,269,975	1,615	1,300
CataTQA(ours)	68	11.62	9,021,397	1,190	29,609

In addition, we compared our previous work and found that we used more data dimensions and generated more practical problems than our previous work. Table 2 shows the detailed statistical information of CataTQA.

3.3 Tools Details

CataTQA has systematically categorized thirteen specialized tools into four primary classes, based on characteristic features of catalysis domain data and problem typologies. The architectural design of these tools is implemented as follows:

Annotation Extraction: We implemented an annotation extraction framework consisting of three specialized tools to facilitate LLM reasoning: 1) a table data reader, 2) A labeled data extractor, and 3) a field information extractor. This toolset was specifically designed to provide structured annotation data to large language models for enhanced reasoning capabilities.

Code Generation: To evaluate the capability of LLMs in generating executable code and answering queries based on given table conditions, we developed two specialized tools: 1) a code generation module and 2) a code execution module.

Question Condition Extraction: Condition analysis is essential for accurate answers. We use four components to systematically derive problem conditions. 1) table sorting, 2) column name analysis, 3) condition extraction, and 4) entity analysis.

Guide Answer Generation: To guide answer generation while accounting for LLMs input constraints, we employ an LLM-driven tool-calling framework comprising four specialized modules: 1) data filtering, 2) data calculation, 3) tool selection, and 4) answer retrieval.

3.4 Expert-Guided Question Generation

The aim of the question generation stage is to create questions answerable using tools in the reference corpus. There are two common approaches: human experts crafting questions based on the corpus, and LLM generating relevant questions. However, both approaches have drawbacks. Human-generated questions demand extensive manpower and time, and lack scalability. LLM-generated questions may be unanswerable, contain fabricated content, or be too simplistic, relying solely on

[1] https://huggingface.co/datasets/kg4sci/CataTQA_Metadata

[2] https://huggingface.co/datasets/kg4sci/CataTQA

the LLM's internal knowledge instead of external tools. This problem is especially prominent in scientific areas like catalysis and medicine.

To address these challenges, we introduce a hybrid question generation approach combining human guidance with LLMs, using structured templates to integrate human input and automated LLMs generation Carpuat et al. (2022); Zhang et al. (2022).Our framework leverages LLMs' natural language understanding and expert-annotated table metadata in prompts to generate high-quality questions from tabular data. LLMs not only create meaningful problem templates aligned with table structures but also generate column names for template variables,enhancing practical question creation and experimental design.Expert verification ensures data quality in the final generated questions.

3.5 Answer Generation

To verify the impact of different question types on table Q&A, we distinguished four types of questions: Cell Query, Fact Judgment, Data Filtering, and Numerical Calculation. The question generation process in CataTQA follows a structured methodology based on expert-annotated table data and supplementary table information. We developed a standardized prompt template (Appendix C.2) that incorporates both the annotated table metadata and predefined question types. Using the Deepseek-671B DeepSeek-AI (2024) model, we generated diverse question templates along with their corresponding required field variables. All generated question templates underwent rigorous expert review and validation. Through programmed template variable instantiation, we systematically converted these templates into functional table-based questions. To ensure answer accuracy, we implemented specialized programming routines for each question type that automatically retrieve correct answers by processing the template variables.

This framework provides precise ground truth labels for all sub-tasks. While CataTQA's main goal is to test LLMs' question-answering ability, we also evaluate their performance in various aspects (Fig. 3), including code generation accuracy and execution results.

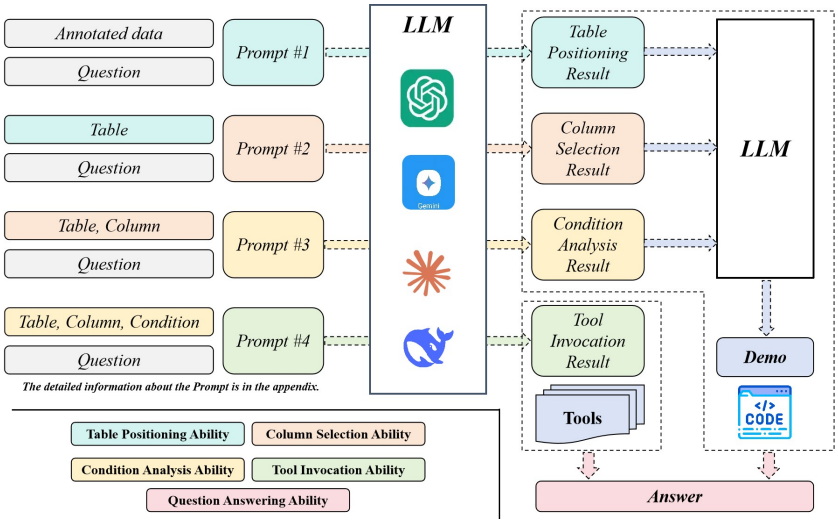

Figure 3: The evaluation process of CataTQA. We have focused on evaluating five capabilities of CataTQA, including table positioning ability, column selection ability, condition analysis ability, tool invocation ability and question answering ability.

4 Experimental Setup

We experimented with four proprietary LLMs on CataTQA, GPT-4o (gpt-4o-2024-11-20) Brown et al. (2020), DeepSeek V3 (deepseek-v3-250324), Claude-3 (claude-3-haiku-20240307) and Gemini-2.5 (gemini-2.5-flash-preview-04-17). For all experiments, we used the same hyperparameters and perform 0-shot prompting via the APIs. The prompts are included in Appendix C.7. We evaluated the LLMs under two different prompting frameworks:

Direct Prompting: Given a task input, LLM is directly prompted to generate the corresponding result. We provide the corresponding prompt instructions for the four sub-tasks in Appendix C.3 to C.6. To ensure experimental fairness, we employed random sampling to select 2,000 questions per problem type, thereby maintaining consistent data conditions across all evaluated models.

Code Generation: Given a task input, LLM is prompted to generate the appropriate program in one pass. We use this framework to evaluate the basic code generation capabilities of each LLM, which is provided in Appendix C.7.For the code generation experiment, we randomly sampled 200 questions from the original set of 2,000 questions used in Direct Prompting.

We evaluated the performance of LLMs on CataTQA in four types of predefined tasks. The CataTQA differs from other benchmarks with a unique set of research challenges. We decomposed the scientific question answering process of LLMs for multiple sources into five key phases: Table Positioning Ability, Column Selection, Condition Analysis, Tool Invocation Ability and Q&A Ability. Following this pipeline, we evaluated the performance of all methods regarding accuracy scores for each sub-task and the final question answering task. The results are summarized in Table 3and 4. To evaluate which sub-task contributes the most, we conducted a comparative study by independently removing all antecedent tasks for each sub-task. For example, in the case of condition analysis, we provide correct table positioning and column selection to individually compare the ability of all models to analyze answer conditions.

5 Results and Analysis

5.1 Main Results

Table 3 presents a comparative analysis of the performance of four LLMs (GPT, Deepseek, Claude, Gemini) across five different question types and multiple evaluation metrics. In general, Gemini appears to be the top-performing model across most question types and evaluation metrics, while other models like GPT, Deepseek, and Claude also show varying levels of competence in different aspects. Detailed analysis will be described in the discussion section.

Table 3: Performance of current LLMs.

Question Type	Model	Table Positioning Ability			Column Selection		Condition Analysis		Tool Invocation	Q&A Ability		
		ACC	ACC@3	ACC@5	ACC	F1	ACC	F1	ACC	ACC	ACC@3	ACC@5
Cell Query	gpt-4o-2024-11-20	0.46	0.64	0.71	0.92	0.95	0.91	0.94	0.27	0.09	0.14	0.15
	deepseek-v3-250324	0.44	0.61	0.70	0.84	0.90	0.93	**0.96**	0.35	0.10	0.15	0.17
	claude-3-haiku-20240307	0.38	0.57	0.66	0.86	0.90	**0.94**	0.95	0.39	0.08	0.14	0.15
	gemini-2.5-flash-preview-04-17	**0.50**	**0.69**	**0.77**	**0.95**	**0.96**	0.85	0.91	**0.50**	**0.18**	**0.26**	**0.29**
Fact Judgment	gpt-4o-2024-11-20	0.43	0.63	0.70	0.94	0.95	0.80	0.88	0.19	0.03	0.06	0.06
	deepseek-v3-250324	0.44	0.63	0.72	**0.95**	0.94	**0.84**	**0.90**	0.14	0.03	0.04	0.05
	claude-3-haiku-20240307	0.38	0.56	0.65	0.91	0.92	0.80	0.88	**0.27**	0.04	0.06	**0.09**
	gemini-2.5-flash-preview-04-17	**0.50**	**0.71**	**0.78**	**0.95**	**0.96**	0.75	0.85	0.25	**0.06**	**0.09**	**0.09**
Data Filtering	gpt-4o-2024-11-20	0.53	0.73	0.78	0.85	0.91	**0.72**	**0.83**	**0.96**	**0.18**	**0.22**	0.25
	deepseek-v3-250324	0.51	0.71	0.77	**0.93**	**0.95**	0.61	0.75	**0.96**	**0.18**	**0.22**	**0.26**
	claude-3-haiku-20240307	0.49	0.68	0.72	0.78	0.86	0.61	0.75	0.94	0.12	0.17	0.18
	gemini-2.5-flash-preview-04-17	**0.54**	**0.74**	**0.84**	0.92	**0.95**	0.37	0.54	**0.96**	0.12	0.17	0.21
Numerical Calculation	gpt-4o-2024-11-20	0.44	0.68	0.73	0.94	0.96	**0.79**	**0.88**	0.44	0.04	0.09	0.09
	deepseek-v3-250324	0.37	0.62	0.72	**0.95**	**0.97**	0.63	0.77	0.43	0.04	0.10	**0.12**
	claude-3-haiku-20240307	0.33	0.56	0.65	0.91	0.94	0.58	0.73	0.07	0.002	0.006	0.008
	gemini-2.5-flash-preview-04-17	**0.47**	**0.72**	**0.81**	0.94	0.96	0.33	**0.50**	0.50	**0.06**	**0.11**	**0.12**

We evaluated the answering ability of the LLM in two ways. The first ability is to assess whether the LLM accurately invokes the correct tool and fills in the correct fields and conditions (see Table 3); the second ability is to directly ask the LLM to generate relevant code, and then determine whether the code can be executed and whether the execution result is accurate. Table 4 shows the evaluation results of the LLM directly generating code to answer questions.

Table 4: The performance of generating code and obtaining answers.

Question Type	Assessment Task	gpt-4o 2024-11-20	deepseek-v3 250324	claude-3 haiku-20240307	gemini-2.5 flash-preview-04-17
Cell Query	Code Executable	0.64	0.70	0.63	**0.84**
	Correct Answer	**0.61**	0.59	0.60	0.52
Fact Judgment	Code Executable	0.81	0.79	0.55	**0.97**
	Correct Answer	0.47	**0.52**	0.35	0.22
Data Filtering	Code Executable	0.91	**0.94**	0.83	0.87
	Correct Answer	**0.67**	0.64	0.52	0.47
Numerical Calculation	Code Executable	**0.87**	0.81	0.72	0.82
	Correct Answer	**0.41**	0.38	0.35	0.31

5.2 Table Positioning Ability

From the Table 3 and Fig. 4(a), we draw the following observations: (1) In categories such as Electrocatalysis, Magnetic materials, and Perovskite, LLMs can accurately locate the correct background knowledge data. However, in the "Others" category, LLMs are more likely to mis-locate data from other fields. This may be because the tables in this category have low specificity, causing interference to the LLMs. (2) Regardless of the type of question, the data-location capabilities of LLMs do not vary significantly. This indicates that LLMs consider the content of questions more than the question categories when selecting data. (3) Overall, Gemini demonstrates the best data-location capabilities, while Claude performs poorly. As shown in Fig. 6 in Appendix, alluvial plot illustrates the table positing ability of different LLMs.

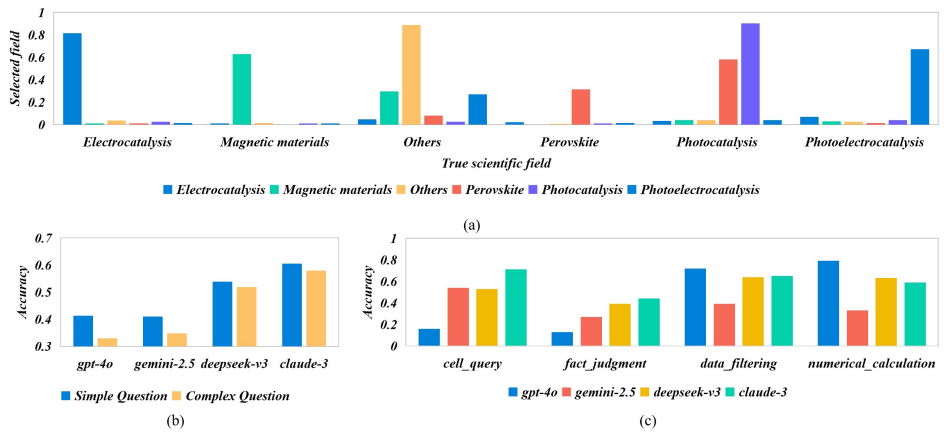

Figure 4: (a) Statistical Analysis of the Table Positioning Ability of LLMs, (b) Comparison of condition analysis accuracy of different LLMs on questions with different difficulty levels, (c) Comparison of condition analysis accuracy of different LLMs on questions with different type. capabilities

5.3 Column Selection Ability&Condition Analysis Ability

Table 3 shows that regardless of which LLM it is, its performance in field selection is superior to other capabilities, and among these models, Gemini performs the best. It can also be easily inferred that question difficulty has little impact, as the process relies more on semantic information than complex reasoning.

As shown Fig. 4(b) & (c), in n terms of conditional analysis capabilities, it can be observed that the performance of Claude-3 is better than that of other models, which may be attributed to its strong logical reasoning ability. Moreover, it can be seen that the difficulty level of the questions has a certain impact on this ability. This is mainly because as the complexity of the questions increases,

the number of conditions also increases accordingly, posing challenges to LLMs. And the type of question also affects the accuracy of conditional judgment. For most LLMs, in problems involving numerical calculations and data filtering, they have demonstrated relatively excellent results in conditional analysis capabilities.Finally, we also conducted experiments on the scope of tool interaction, and the experimental results are presented in D.2.

5.4 Tool Invocation Ability

From the results, it can be seen that in the type of questions related to Data Filtering, all large models can effectively invoke the corresponding tools. However, in other types of questions, the performance of large models is not satisfactory. Fig. 5 shows the confusion matrix of the tool invocation results of large models. All LLMs tend to confuse the Data Filtering tool with other tools. This may be due to the inductive summarization ability of large models, resulting in poor performance in questions related to data calculation, situation judgment, etc. Questions of inductive summarization tend to call the Data Filtering tool rather than other types of tools. Except for Claude-3, other models can effectively use the calculate data tool to answer questions.

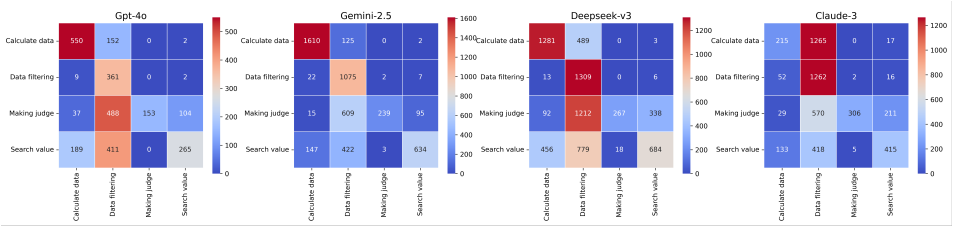

Figure 5: Comparison of Confusion Matrices of Tool Invocation Capabilities from Different LLMs

6 Conclusion and Limitations

As one of the most intuitive representations of scientific data, tabular data contains a wealth of latent domain knowledge. Making full use of open domain datasets can assist scientists in better exploring the development context of the field and achieving scientific discoveries. Additionally, it facilitates the development of specialized chatbots that can generate answers based on table content. Therefore, we developed CataTQA, designed to evaluate LLMs' ability to accurately understand background knowledge in scientific domains, precisely locate valid information within vast domain datasets, and invoke external tools to solve complex problems. The development process of CataTQA is highly generalizable, enabling seamless migration to other scientific disciplines. We conducted extensive tests on CataTQA using GPT, DeepSeek, Claude and Gemini, and the results revealed that even the strongest model achieved limited performance on the hard questions of CataTQA.

However, CataTQA has significant limitations. Firstly, it currently doesn't support cross-table associative retrieval(We have made some attempts in our research in Appendix.D.1.). As a result, the dataset can hardly simulate the complex problems in real scientific research, and it may underestimate the models' ability to handle associated data. Secondly, CataTQA is mainly focused on the catalyst field at present. To enhance its comprehensiveness, it's necessary to expand the dataset to cover more disciplines like astronomy, genomics, etc. This tests models' ability to handle diverse data and scientific problems, enabling a more comprehensive evaluation of generalization.Finally,recent advancements in scientific annotation tools have significantly enhanced both the quality and efficiency of metadata annotation. Representative systems such as Autodive Du et al. (2023) demonstrate this progress by enabling automated annotation of tabular data within PDF-formatted scientific papers, thereby facilitating rapid extraction of scientific table data. Our approach currently lacks integration with established annotation tools, which could better reflect real-world scenarios.In the future, efforts will be made to address these shortcomings and strengthen the role of CataTQA as a benchmark for evaluating large language models in scientific applications.

REFERENCES

Bishwaranjan Bhattacharjee, Aashka Trivedi, Masayasu Muraoka, Muthukumaran Ramasubramanian, Takuma Udagawa, Iksha Gurung, Nishan Pantha, Rong Zhang, Bharath Dandala, Rahul Ramachandran, et al. Indus: Effective and efficient language models for scientific applications. *arXiv preprint arXiv:2405.10725*, 2024.

Abeba Birhane, Atoosa Kasirzadeh, David Leslie, and Sandra Wachter. Science in the age of large language models. *Nature Reviews Physics*, 5(5):277–280, 2023.

Ben Bogin, Kejuan Yang, Shashank Gupta, Kyle Richardson, Erin Bransom, Peter Clark, Ashish Sabharwal, and Tushar Khot. Super: Evaluating agents on setting up and executing tasks from research repositories. *arXiv preprint arXiv:2409.07440*, 2024.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sébastien Bubeck, Varun Chadrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.). *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States, July 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.naacl-main.0/.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, et al. Finqa: A dataset of numerical reasoning over financial data. *arXiv preprint arXiv:2109.00122*, 2021.

Subramanian Chidambaram, Li Erran Li, Min Bai, Xiaopeng Li, Kaixiang Lin, Xiong Zhou, and Alex C Williams. Socratic human feedback (sohf): Expert steering strategies for llm code generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15491–15502, 2024.

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

José Cassio dos Santos Junior, Rachel Hu, Richard Song, and Yunfei Bai. Domain-driven llm development: Insights into rag and fine-tuning practices. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6416–6417, 2024.

Yi Du, Ludi Wang, Mengyi Huang, Dongze Song, Wenjuan Cui, and Yuanchun Zhou. Autodive: An integrated onsite scientific literature annotation tool. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 76–85, 2023.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pp. 10764–10799. PMLR, 2023.

Jorge Osés Grijalba, L Alfonso Urena Lopez, Eugenio Martínez-Cámara, and Jose Camacho-Collados. Question answering over tabular data with databench: A large-scale empirical evaluation of llms. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 13471–13488, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 1(2):4, 2022.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Yeonghun Kang and Jihan Kim. Chatmof: an artificial intelligence system for predicting and generating metal-organic frameworks using large language models. *Nature communications*, 15(1): 4705, 2024.

Julia Koblitz, Sabine Eva Will, S Alexander Riemer, Thomas Ulas, Meina Neumann-Schaal, and Dietmar Schomburg. The metano modeling toolbox mmtb: an intuitive, web-based toolbox introduced by two use cases. *Metabolites*, 11(2):113, 2021.

Agustinus Kristiadi, Felix Strieth-Kalthoff, Marta Skreta, Pascal Poupart, Alán Aspuru-Guzik, and Geoff Pleiss. A sober look at llms for material discovery: Are they actually good for bayesian optimization over molecules? *arXiv preprint arXiv:2402.05015*, 2024.

Bernadette Lamb, Jonathan Herskovitz, Marta Jonson, Harlan Sayles, and Faruq Pradhan. S2185 use of large language model (llm) chatbots for the generation of specialized gastrointestinal diet meal plans: A pilot study. *Official journal of the American College of Gastroenterology— ACG*, 119(10S):S1562, 2024.

Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Woncheol Shin, Seongjun Yang, Minjoon Seo, Jong-Yeup Kim, and Edward Choi. Ehrsql: A practical text-to-sql benchmark for electronic health records. *Advances in Neural Information Processing Systems*, 35:15589–15601, 2022.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36:43447–43478, 2023.

Aman Madaan and Amir Yazdanbakhsh. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*, 2022.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Investigating the limitations of transformers with simple arithmetic tasks. *arXiv preprint arXiv:2102.13019*, 2021.

Karen O'Leary. Llms get a medical education. *Nature Medicine*, 2023.

Maciej P Polak and Dane Morgan. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications*, 15(1):1569, 2024.

Jing Qian, Hong Wang, Zekun Li, Shiyang Li, and Xifeng Yan. Limitations of language models in arithmetic and symbolic induction. *arXiv preprint arXiv:2208.05051*, 2022.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Sohini Roychowdhury, Marko Krema, Anvar Mahammad, Brian Moore, Arijit Mukherjee, and Punit Prakashchandra. Eratta: Extreme rag for enterprise-table to answers with large language models. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 4605–4610. IEEE, 2024.

Mobashir Sadat, Zhengyu Zhou, Lukas Lange, Jun Araki, Arsalan Gundroo, Bingqing Wang, Rakesh R Menon, Md Rizwan Parvez, and Zhe Feng. Delucionqa: Detecting hallucinations in domain-specific question answering. *arXiv preprint arXiv:2312.05200*, 2023.

Abdullahi Saka, Ridwan Taiwo, Nurudeen Saka, Babatunde Abiodun Salami, Saheed Ajayi, Kabiru Akande, and Hadi Kazemi. Gpt models in construction industry: Opportunities, limitations, and a use case validation. *Developments in the Built Environment*, 17:100300, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

Ethan Waisberg, Joshua Ong, Mouayad Masalkhi, and Andrew G Lee. Large language model (llm)-driven chatbots for neuro-ophthalmic medical education. *Eye*, 38(4):639–641, 2024.

Xingyao Wang, Sha Li, and Heng Ji. Code4struct: Code generation for few-shot structured prediction from natural language. *arXiv preprint arXiv:2210.12810*, 3, 2022.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.

Yiran Wu, Feiran Jia, Shaokun Zhang, Hangyu Li, Erkang Zhu, Yue Wang, Yin Tat Lee, Richard Peng, Qingyun Wu, and Chi Wang. Mathchat: Converse to tackle challenging math problems with llm agents. *arXiv preprint arXiv:2306.01337*, 2023.

Qiantong Xu, Fenglu Hong, Bo Li, Changran Hu, Zhengyu Chen, and Jian Zhang. On the tool manipulation capability of open-sourced large language models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Fangkai Yang, Pu Zhao, Zezhong Wang, Lu Wang, Jue Zhang, Mohit Garg, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Empower large language model to perform better on industrial domain-specific question answering. *arXiv preprint arXiv:2305.11541*, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Guanghui Ye, Huan Zhao, Zixing Zhang, and Zhihua Jiang. Unide: A multi-level and low-resource framework for automatic dialogue evaluation via llm-based data augmentation and multitask learning. *Information Processing & Management*, 62(3):104035, 2025.

Mohd Zaki, NM Anoop Krishnan, et al. Mascqa: investigating materials science knowledge of large language models. *Digital Discovery*, 3(2):313–327, 2024.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, et al. Chemllm: A chemical large language model. *arXiv preprint arXiv:2402.06852*, 2024a.

Huan Zhang, Yu Song, Ziyu Hou, Santiago Miret, and Bang Liu. Honeycomb: A flexible llm-based agent system for materials science. *arXiv preprint arXiv:2409.00135*, 2024b.

Rongzhi Zhang, Yue Yu, Pranav Shetty, Le Song, and Chao Zhang. Prboost: Prompt-based rule discovery and boosting for interactive weakly-supervised learning. *arXiv preprint arXiv:2203.09735*, 2022.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. *arXiv preprint arXiv:2105.07624*, 2021.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36: 50117–50143, 2023.

Andreas Ziegler, David Joseph, Thomas Gossard, Emil Moldovan, and Andreas Zell. Biasbench: A reproducible benchmark for tuning the biases of event cameras. *arXiv preprint arXiv:2504.18235*, 2025.

A Data Source

A.1 Data Source

We have retrieved data from multiple open-source databases. The following lists the top 9 categories of data with large volumes. For the complete data sources, please refer to the tabular metadata in our GitHub repository.

- **Catalyst Acquisition by Data Science(CADS)**[1]: An innovative web-based integrated catalyst informatics platform, Catalyst Acquisition by Data Science (CADS), is developed for use towards the discovery and design of catalysts.
- **Catalytic Material Database(CMD)**[2]: CMD contains material composition, properties, reactions, products and other information.
- **Catalyst Hub**[3]: A featured database for surface reactions contains more than 100,000 chemisorption and reaction energies obtained from electronic structure calculations, and is continuously being updated with new datasets.
- **Crystallography Open Database(COD)**[4]: An open-access collection of crystal structures of organic, inorganic, metal-organic compounds and minerals.
- **Materials Project**[5]: The Materials Project provides computed information on known and predicted materials as well as powerful analysis tools to inspire and design novel materials.
- **2DMatPedia dataset**[6]: DMatPedia dataset is a collection of 2D materials, contains 6351 materials.
- **Alexandria_DB PBE 3D**[7]:A dataset of 2.5m+ stable and metastable materials calculated with the PBE functional.
- **OQMD-3D dataset**[8]: The OQMD is a database of DFT calculated thermodynamic and structural properties of 1,226,781 materials, created in Chris Wolverton's group at Northwestern University.
- **Associated Data of Papers**: We have also obtained the data related to the paper. For the specific information of the paper, please refer to the project on GitHub.

[1] https://cads.eng.hokudai.ac.jp/
[2] http://cmd.us.edu.pl/catalog/
[3] http://www.catalysthub.net/
[4] http://www.crystallography.net/cod/
[5] https://next-gen.materialsproject.org/
[6] http://www.2dmatpedia.org/
[7] https://alexandria.icams.rub.de/
[8] https://www.oqmd.org/download/

A.2 ANNOTATION INSTANCE

We annotated each dataset as shown below.

---

tabular name: table1

tabular description: This table evaluates Mo-doped BaTiO3 photocatalysts for hydrogen production, correlating doping levels with bandgap (Eg) and activity (RH2). It identifies optimal Mo content (2%) for peak efficiency, aiding material design for enhanced solar-driven water splitting. Contributions include optimizing dopant ratios to balance light absorption and charge separation.

field description: 1.[Molecular formula] - [elemental composition of compounds]
2.[RH2(μmol h-1 g-1)] - [Hydrogen production rate]
3.[Eg(eV)] - [Band gap energy]
4.[Preparation method] - [Material synthesis technique]
5.[Calcination temperature(K)] - [Heating temperature during synthesis]
6.[Calcination time(h)] - [Heating duration during synthesis]
7.[Light intensity(W)] - [Light source power used]
8.[Reaction solution] - [Chemical solution in reaction]
9.[Co-catalyst] - [Catalyst used with photocatalyst]
10.[Photocatalyst dose(g L-1)] - [Photocatalyst concentration used]
11.[Ref] - [Reference source citation]

A.3 DATA INSTANCE

---

question: Find the partial pressure of argon (Par) when the methane partial pressure (Pch4) is 0.3, oxygen partial pressure (Po2) is 0.06, and temperature is 850°C.

refer_dataset: table28

column names: {*Pch4, Po2, Temperature, Par*}

condition_column: {*Pch4, Po2, Temperature*}

answer_column: {*Par*}

condition: {Pch4: 0.3, Po2: 0.06, Temperature: 850}

tool: search_value

answer: Par: 0.6

level: ***complex***

question description: In a tabular data structure, locate the cells that meet the requirements.

refer_template: Find the partial pressure of argon (Par) when the methane partial pressure (Pch4) is {}, oxygen partial pressure (Po2) is {}, and temperature is {}°C.

B Question Templates

B.1 Cell Query Templates

Cell Query Templates

- **Find the Pauling electronegativity of element B (χ_{pb}) in {}.**
- **How much is the molar mass (M) of the compound named {}?**
- **What license type applies to the paper authored by {} and last updated on {}?**

B.2 Fact Judgment Templates

Fact Judgment Templates

- **Is a loading of {} g/l combined with a reaction medium of {}?**
- **Does any material with {} as the primary B site element have a tertiary A site element of {}?**
- **Is there a perovskite with PBE and HSE lattice constants of {} and {}, respectively, and a refractive index of {}?**

B.3 Data Filtering Templates

Data Filtering Templates

- **For support material {}, which M1 elements paired with M2 {} yield H2 over {}% and O2 conversion under {}%?**
- **Find compounds in sub_family with axial distortion exceeding {} and slope_CSM under {}.**
- **Show materials with phase separation energy ¡ {} and atoms matching lattice type {}?**

B.4 Numerical Calculation Templates

Numerical Calculation Templates

- **What is the minimum e_hull value for materials with bulk modulus exceeding {}?**
- **Calculate the average PBE0 open-circuit voltage for entries with PBE0 LUMO energy level greater than {}.**
- **What is the highest First Ionization Energy of B (I1b) observed in materials with EAa greater than {}, EAb less than {}, and ρa exceeding {}?**

C PROMPTS

C.1 PROMPTS FOR TABLE ANNOTATION

<Table Annotation>Prompt

Annotate the table data, summarize the main problems that this table can solve and its contributions based on the content of the table data.
Table data:
{dataset}
Output requirements:
1. Summarize the role of data and avoid discussing a single column or row of data
2. Output is limited to 50 words or less

C.2 PROMPTS FOR TEMPLATE QUESTIONS GENERATION

<Template Questions Generation>Prompt

Annotate the table data, summarize the main problems that this table can solve and its contributions based on the content of the table data.
Please generate questions according to the following rules:
1. Requirements to be met:
- template questions type: {question_description}
- number of columns required to obtain answers: at least two columns
- level: The level of the template questions is differentiated according to the number of columns used.
Including two levels of simple and complex.
2. Example:
Input:
- table description:{example_tabular_description}
- [column names] - [description]:{example_field_description}
Output:{example}
3. output format:
- Mark the level of each question.At least ten questions per level.
- Mark the column names that need to be used to answer this question template.
- Use"{{}}" for template variables.The template variable must be one of the columns of the table.
- Use of multiple sentence structures.Questions need to be phrased in a way that is easy to understand.
Use the information in the table below to generate template questions according to the above rules:
Input:
- table description:{tabular_description}
- [column names] - [description]:{field_description}

C.3 PROMPTS FOR TABLE ORDER

<Table Order>Prompt

Please analyze the relevance of the table according to the problem.
question:{question}
Available tables and table descriptions:
{table_desc}
Please sort the tables by relevance from high to low, give the first five possible tables, and directly return the table name list.
For example: ["table1", "table2"]

C.4 Prompts for Column Selection

<Column Selection>Prompt

Only provide the column names required to answer the question:
question: {question}
{table} information: [column]-[column name explanation]
{table_field}
directly return to the column name list, for example: ['col1','col2']

C.5 Prompts for Condition Extraction

<Condition Extraction>Prompt

Please extract the query criteria from the question and return the results according to the table structure:
question: {question}
table information:
column name: [{tar}]
Output requirements:
1. Return to dictionary format, with the key being the column name and the value being the query condition
2. The values corresponding to all column names must be output. The output not found in the problem is ".
3. Extract and preserve comparison symbols (¿,¡,=, etc.)
4. example:"column1":"¿50", "column2":"Liming"
Please return the JSON dictionary directly without including any other content

C.6 Prompts for Tool Invocation

<Tool Invocation>Prompt

Please use the tools needed to answer the questions according to the question analysis. You need to specify the calculated column name when you need to perform calculation, but you do not need to specify it when you are performing table lookup.
Question: {question}
Description of available tools:
{tool_desc}
Please return the tool and column name to be used directly. For example: {{"tool":"","colnum name":""}}

C.7 PROMPTS FOR GENERATING CODE

<Generating Code>Prompt

Generate python code based on the question and given conditions, index CSV file data, and answer the question.
Generate a function named "get_answer"(No parameters required). The function must use the "return" keyword to return the variable "answer", which is the answer to the question.
question: {question['question']}
refer_dataset: {question['refer_dataset']}
column names: {question['column names']}
condition: {question['condition']}
Code must be used in markdown format("python").
Do not return redundant content.The returned results must be saved in the "answer" variable.

D SUPPLEMENTARY EXPERIMENTS

D.1 CROSS-TABLE REASONING

In the field of materials science, there is a need to perform reasoning across table information to address more complex questions and derive answers. In this paper, while constructing a benchmark for single tables, we also attempted to build a multi-table association reasoning task. We migrated the current pipeline to multi-table task reasoning, using metadata and table descriptions as additional inputs to prompt LLMs to generate cross-table task question templates. In total, we generated 426 cross-table questions and invited domain experts to evaluate the rationality and complexity of these questions. However, after expert evaluation, only 37.5% of the questions were satisfactory. Considering the value and accuracy of the benchmark comprehensively, we did not include them in CataTQA, and we also described this part in the "limitation" section of the paper. In the future, we will continue to update the pipeline and further expand the evaluation data according to the characteristics of cross-table reasoning tasks.

Nevertheless, as can be seen from the evaluation results in this paper, even in reasoning for single tables, the performance of mainstream LLMs still has significant room for improvement. Therefore, we believe that this work can serve as a new benchmark for evaluating whether large models can accurately understand information in the catalytic field, locate tables and conditions, and invoke the correct tools.

Cross-Table Reasoning Examples:

- **Question 1: For materials with a molecular formula of {table1.Molecular formula} and a bandgap energy of {table2.Eg(eV)}, what is the co-catalyst used and the first ionization energy of element B?**
  Annotation from expert: The two tables differ in their research objects and data structures, so directly correlating molecular formulas with band gap energies may lead to data confusion. In addition, molecular formulas and band gap energies belong to the basic properties of materials, while Cocatalysts and the ionization energy of element B fall under process and elemental attributes. There is no direct scientific or database logical connection between these two categories.
- **Question 2: What is the effect of the light type {table4.Light Type} on the bandgap energy {table2.Eg(eV)} and the density of element A {table2.a} in materials with a primary element A of {table4.A}?**
  Annotation from expert: This question has a muddled causal relationship. The type of light (table4.Light Type) is an external experimental condition, while the band gap energy (table2.Eg(eV)) and element density (table2.a) are intrinsic properties of the material. The type of light does not directly affect the band gap or element density of the material, and there is no scientific causal relationship between them.

- **Question 3: What is the correlation between the publication year {table12.Year} of studies on catalysts with surface strain {table12.Surface Strain} and the number of times they were cited {table13.Cited_Time}?**

  Annotation from expert: The correlation between the different variables proposed in this question lacks scientific significance. The publication year is a temporal attribute of a paper, while surface strain is a physical property of materials; there is no direct scientific connection between the two. The number of citations a paper receives is mainly influenced by factors such as research impact and field hotspots, and has no direct causal relationship with the surface strain of materials.

D.2 Scope of Tool Interaction

We have evaluated and analyzed the accuracy or failure modes of tool components to conduct a more comprehensive assessment of the capabilities of large language models.

For the tasks of Table Positioning Ability and Column Selection, LLM can output in the required format without discovering any other faults, due to our limited output constraints. LLM has excellent ability to complete the answers in these two tasks.For the tasks of Condition Analysis and Tool Invocation, we require LLM to output a more rigorous JSON format.Our diagnostic analysis revealed that 31% of failures stemmed exclusively from JSON formatting errors in the LLM's output. While this represents an improvement over previous format requirements, we acknowledge this remains a significant error source.

For our primary research focus on assessing LLMs' tabular reasoning capabilities. We intentionally excluded formatting errors from the final accuracy calculations.Only properly formatted outputs were evaluated for task performance.This approach ensures our metrics reflect the LLM's core table reasoning abilities rather than format compliance.This methodological choice aligns with our research objective to isolate and measure the model's table processing competencies, while recognizing formatting as an orthogonal challenge.

In the QA Ability task, we conducted a detailed code check and, based on our task division, LLM deduced the correct preconditions. By calling relevant tools, we were able to almost complete the correct answer retrieval.

We also further analyzed the reasons for the incorrect invocation of Data Filtering tool, and the results of the analysis are as follows:

As can be seen from Fig.5, when facing all types of problems, LLMs often incorrectly call data filtering tools to answer the questions. We further analyzed the possible causes for different types of problems.

For questions that require calling the calculate_data tool for answers, LLMs often incorrectly invoke the data_filtering tool because they fail to understand the calculation instructions in the questions (such as minimum, maximum, average, etc.). For example, regarding the question "Determine the minimum Fermi energy level (efermi) for materials where SCF valence band maximum (scf_vbm) is greater than 5.176." the large model failed to accurately comprehend "minimum efermi" when analyzing the required conditions, and only generated the condition information "llm_condition": "scf_vbm": ">5.176", "efermi": "". Therefore, it chose to call the data_filtering tool to filter the table data.

For questions that require invoking the makin_judge tool for answers, LLMs incorrectly call the data_filtering tool because they fail to fully understand that the instructions demand a judgmental response of "yes or no". This may be due to the fact that data-related information in the question's syntax is positioned later, thereby causing the misunderstanding. For instance, in the question "Is there a material where M06 method reports both HOMO of -0.235, LUMO of -0.084, and Scharber efficiency of 0.00034537252143?", the LLMs focused heavily on the information about "HOMO", "LUMO", and "Scharber efficiency" while ignoring the question's intent "Is there a material", leading to the invocation of the data_filtering tool.

In the analysis of tool invocation confusion in questions that require the search_value tool, we found that the issue of incorrect tool calls by large models may lie in the ambiguity in their judgment of column name selection. For example, in the question "What is the chemical formula for the material

containing the element -C-H?", the large model mistakenly identified "containing the element -C-H" as a data filtering condition, and thus invoked the data_filtering tool to answer the question.

It can be seen from the results that current LLMs still have significant room for improvement in the correctness of tool calls in such specialized fields as catalysis. Therefore, we believe that CataTQA can serve as a benchmark for evaluating the application effects of LLMs in practical scientific research fields.

E Other Information

E.1 Verify the origin of the answer

To verify whether LLMs can answer relevant questions based on their own memorization, we conducted supplementary experiments. We directly input the questions into the large model and asked it to provide answers, then evaluated the accuracy of its responses. As can be seen from the examples below, the LLMs can not answer the questions accurately. Therefore, It can prove to a certain extent that this benchmark is mainly used to test the reasoning ability of LLMs rather than their memorization, in the catalytic field.

We use the model 'gpt-4o-2024-11-20' and the following prompt:

<Verify the origin of the answer >Prompt

Please provide the answer directly to the question without giving any explanation.
question:{question}
The output format is: ```json{{"answer": ""}}```

Table 5: Result of verify the origin of the answer.

Question Type	Level	ACC	AVG ACC
Cell Query	simple	0.004	0.002
	complex	0	
Fact Judgment	simple	0.253	0.236
	complex	0.217	
Data Filtering	simple	0	0
	complex	0	
Numerical Calculation	simple	0.034	0.056
	complex	0.094	

The table(Table.5) shows our experimental results.By randomly selecting 500 questions from each category in the dataset we created for LLM to answer directly. In the end, we calculate an average accuracy of 0.002 for Cell_Query, 0.236 for Fact_Judgment, 0.0 for Data Filtering, and 0.056 for Numerical Calculation. Although the accuracy rate is slightly higher on Fact Judge questions, this does not necessarily mean that LLM has answered the questions based on their own knowledge, because for Fact Judge questions, LLM only needs to provide 'true' or 'false' answers, and the accuracy rate of random guessing on this question may reach 50%.However, now it is 23.6% for Fact_Judgment.

Therefore, this experiment proves that the data we used does not overlap with the data pre-trained by LLM.

CataTQA codebase is hosted and version-tracked via GitHub. It will be permanently available under the link `https://github.com/qcui2025/CataTQA`. The download link of all the datasets can be found in the GitHub repository.

CataTQA stands as a community-driven open-source initiative. We are not only firmly committed but also well-resourced to sustain and vigorously advance it in the times ahead. Envisioning its expansion, we aim to incorporate a more diverse range of tasks, tools, and baseline methodologies. Additionally, we warmly welcome external contributors to partake in this endeavor, as their involvement will be pivotal in shaping the future trajectory of CataTQA.

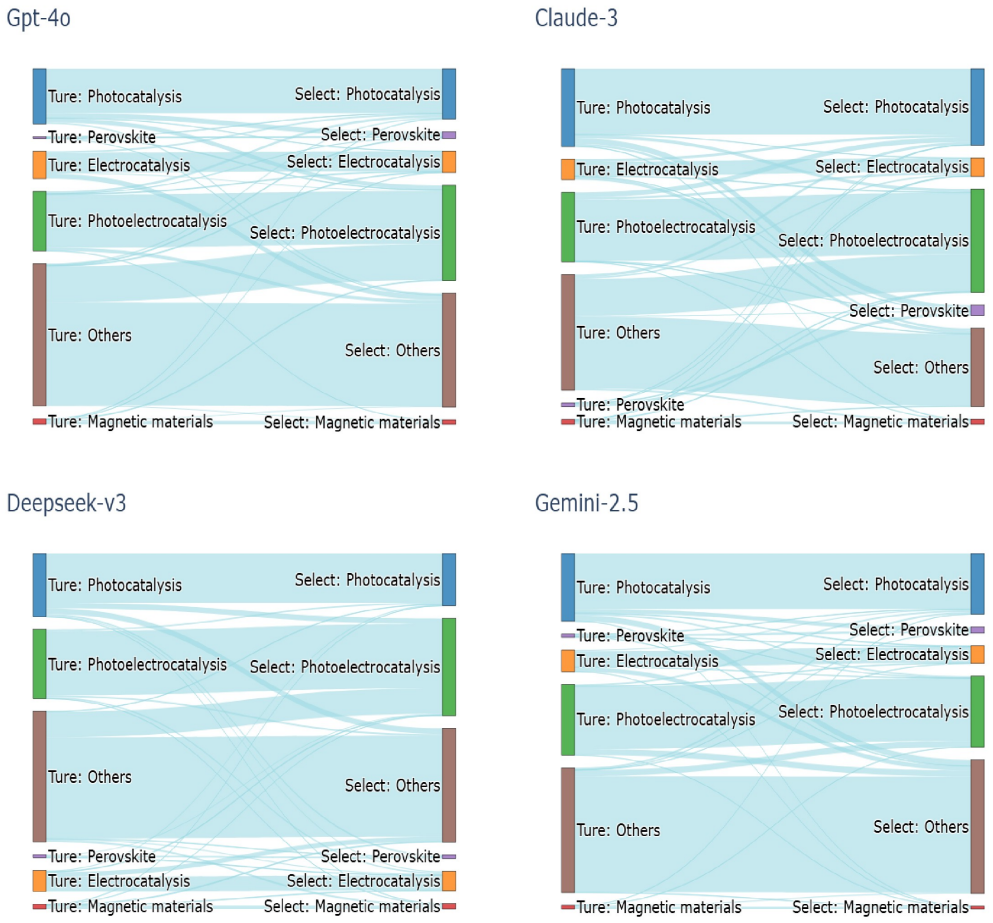

Figure 6: Alluvial plot of the Table Positioning Ability of Large Language Models

We license our work using Apache 2.0. All the datasets will be publicly released through the aforementioned GitHub link.

The authors will bear all responsibility in case of violation of rights.

E.2 Full dataset information

Table 6: Detailed Information of CataTQA Raw Data

Domain	Dataset or paper name	Access url	Index
photo-catalysis	Machine learning aided design of perovskite oxide materials for photocatalytic water splitting	https://www.sciencedirect.com/science/article/pii/S2095495621000644#s0090	table1 table2 table3
	Data mining in photocatalytic water splitting over perovskites literature for higher hydrogen production	https://www.sciencedirect.com/science/article/pii/S0926337318309470#sec0130	table4 table5
	An insight into tetracycline photocatalytic degradation by MOFs using the artificial intelligence technique	https://www.nature.com/articles/s41598-022-10563-8#Sec10	table6
	Analysis of photocatalytic CO2 reduction over MOFs using machine learning	https://pubs.rsc.org/en/content/articlelanding/2024/ta/d3ta07001h	table7
	Data-driven for accelerated design strategy of photocatalytic degradation activity prediction of doped TiO2 photocatalyst	https://www.sciencedirect.com/science/article/pii/S2214714422005700#s0055	table8
	A generalized predictive model for TiO2–Catalyzed photo-degradation rate constants of water contaminants through artificial neural network	https://www.sciencedirect.com/science/article/pii/S0013935120305909	table9

Domain	Dataset or paper name	Access url	Index
	Statistical information review of CO2 photocatalytic reduction via bismuth-based photocatalysts using artificial neural network	https://www.sciencedirect.com/science/article/pii/S1110016824008640?via=ihub	table10
	Accelerated Design for Perovskite-Oxide-Based Photocatalysts Using Machine Learning Techniques	https://www.mdpi.com/1996-1944/17/12/3026	table11
electro-catalysis	Building Blocks for High Performance in Electrocatalytic CO2 Reduction: Materials, Optimization Strategies, and Device Engineering	https://acs.figshare.com/articles/dataset/Building_Blocks_for_High_Performance_in_Electrocatalytic_CO_sub_2_sub_Reduction_Materials_Optimization_Strategies_and_Device_Engineering/5293804?file=9064090	table12
	Unlocking New Insights for Electrocatalyst Design: A Unique Data Science Workflow Leveraging Internet-Sourced Big Data	https://github.com/ruiding-uchicago/InCrEDible-MaT-GO	table13
	Perovskite-based electrocatalyst discovery and design using word embeddings from retrained SciBERT language model	https://github.com/arunm917/Perovskite-based-electrocatalyst-design-and-discovery	table14

Domain	Dataset or paper name	Access url	Index
	Exploring the Composition Space of High-Entropy Alloy Nanoparticles for the Electrocatalytic H2/CO Oxidation with Bayesian Optimization	https://github.com/vamints/Scripts_BayesOpt_PtRuPdRhAu_paper	table15
	High Throughput Discovery of Complex Metal Oxide Electrocatalysts for the Oxygen Reduction Reaction	https://data.caltech.edu/records/1km87-52j70	table16
photo-electro-catalysis	High-thoughput OCM data	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/21010bbe-0a5c-4d12-a5fa-84eea540e4be/	table17
	CatApp Data	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/20de069b-53cf-4310-9090-1738f53231e2/	table18
	Oxidative Coupling of Methane	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/9436f770-a7e2-4e87-989b-c5a9ce2312bf/	table19
	ChemCatChem	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/224dd7ad-7677-4161-b744-a0c796bf5347/	table20
	HTP OCM data obtained with catalysts designed on the basis of heuristics derived from random catalyst data	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/92200ba4-7644-44ca-9801-ed3cc52fc32f/	table21

Domain	Dataset or paper name	Access url	Index
	Perovskite Data	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/f1b42c58-a423-4ec2-8bcf-e66c6470ff7d/	table22
	Random catalyst OCM data by HTE	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/f7e30001-e440-4c1a-be64-ea866b2f77cb/	table23
	Synthesis of Heterogeneous Catalysts in Catalyst Informatics to Bridge Experiment and High-Throughput Calculation	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/f2a6d4f2-91be-48ba-bf13-ffebbd90f6ee/	table24
	Multi-component La2O3-based catalysts in OCM	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/d6347fc1-e4d7-412e-aed5-a8ffa415a703/	table25
	Catalyst Modification in OCM via Manganese Promoter	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/32dbec2c-c3d5-43ec-962a-90dba719bb44/	table26
	Leveraging Machine Learning Engineering to Uncover Insights in Heterogeneous Catalyst Design for Oxidative Coupling of Methane	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/d84c1e22-ceb9-488a-8d45-4c7cf1c603b5/	table27
	Oxidative of Coupling Literature and Highthroughput Data	https://cads.eng.hokudai.ac.jp/datamanagement/datasources/adb27910-d0e5-4a22-9415-580bf597035a/	table28

Domain	Dataset or paper name	Access url	Index
	Catalytic Material Database	http://cmd.us.edu.pl/catalog/	table29 table30
	Catalyst Hub	-	table31
magnetic material	Magnetic Database	https://doi.org/10.15131/shef.data.24008055.v1	table32
	Materials database of Curie and Néel magnetic phase transition temperatures	https://doi.org/10.6084/m9.figshare.5702740.v1	table33
perovskite	Data-driven design of molecular nanomagnets	https://go.uv.es/rosaleny/SIMDAVIS	table34
	Predicting the thermodynamic stability of perovskite oxides using machine learning models	-	table35 table36 table37
others	Crystallography Open Database(COD)	http://www.crystallography.net/cod/	table38
	Alloy synthesis and processing by semi-supervised text mining	https://www.nature.com/articles/s41524-023-01138-w	table39
	A Machine Learning Approach to Zeolite Synthesis Enabled by Automatic Literature Data Extraction	https://github.com/olivettigroup/table_extractor	table40
	ZeoSyn: A Comprehensive Zeolite Synthesis Dataset Enabling Machine-Learning Rationalization of Hydrothermal Parameters	https://github.com/eltonpan/zeosyn_dataset	table41

Domain	Dataset or paper name	Access url	Index
	Unveiling the Potential of AI for Nanomaterial Morphology Prediction	https://github.com/acid-design-lab/Nanomaterial_Morphology_Prediction	table42
	AFLOW-2 CFID dataset	https://doi.org/10.1016/j.commatsci.2012.02.005	table43
	Alexandria_DB PBE 3D all	https://alexandria.icams.rub.de/	table44
	arXiv dataset	https://www.kaggle.com/Cornell-University/arxiv	table45
	CCCBDB dataset	https://cccbdb.nist.gov/	table46
	3D dataset	https://www.nature.com/articles/s41524-020-00440-1	table47
	2D dataset	https://www.nature.com/articles/s41524-020-00440-1	table48
	halide perovskite dataset	https://doi.org/10.1039/D1EE02971A	table49
	hMOF dataset	https://doi.org/10.1021/acs.jpcc.6b08729	table50
	HOPV15 dataset	https://www.nature.com/articles/sdata201686	table51
	Surface property dataset	https://doi.org/10.1039/D4DD00031E	table52
	JARVIS-FF	https://www.nature.com/articles/s41524-020-00440-1	table53
	MEGNET-3D CFID dataset	-	table54
	Materials Project-3D CFID dataset	https://next-gen.materialsproject.org/	table55
	Materials Project-3D CFID dataset 84k	–	table56
	OQMD-3D dataset	https://www.oqmd.org/download/	table57

Domain	Dataset or paper name	Access url	Index
	Polymer genome	https://datadryad.org/dataset/doi:10.5061/dryad.5ht3n	table58
	QETB dataset	https://arxiv.org/abs/2112.11585	table59
	QM9 dataset	https://www.nature.com/articles/sdata201422	table60
	QM9 standardized dataset 130k	–	table61
	QMOF dataset	https://www.cell.com/matter/fulltext/S2590-2385(21)00070-9	table62
	SNUMAT Hybrid functional dataset	https://www.nature.com/articles/s41597-020-00723-8	table63
	SSUB dataset	https://github.com/wolverton-research-group/qmpy	table64
	chem dataset	https://www.nature.com/articles/s41524-018-0085-8	table65
	InterMat dataset	https://doi.org/10.1039/D4DD00031E	table66
	2DMatPedia dataset	http://www.2dmatpedia.org/	table67
	vacancy dataset	https://doi.org/10.1063/5.0135382	table68