On LLM-Based Scientific Inductive Reasoning Beyond Equations

Brian S. Lin[1*] **Jiaxin Yuan**[2*] **Zihan Zhou**[3*] **Shouli Wang**[4*] **Shuo Wang**[1†]
Cunliang Kong[1] **Qi Shi**[1] **Yuxuan Li**[1] **Liner Yang**[2†] **Zhiyuan Liu**[1†] **Maosong Sun**[1]
[1]Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center, Tsinghua University
Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University
[2]Beijing Language and Culture University
[3]Xiamen University [4]Harbin Institute of Technology
caish25@mails.tsinghua.edu.cn

Abstract

As large language models (LLMs) increasingly exhibit human-like capabilities, a fundamental question emerges: How can we enable LLMs to learn the underlying patterns from limited examples in entirely novel environments and apply them effectively? This question is central to the ability of LLMs in inductive reasoning. Existing research on LLM-based inductive reasoning can be broadly categorized based on whether the underlying rules are expressible via explicit mathematical equations. However, many recent studies in the beyond-equations category have emphasized rule design without grounding them in specific scenarios. Inspired by the parallels between inductive reasoning and human scientific discovery, we propose the task of LLM-Based Scientific Inductive Reasoning Beyond Equations and introduce a new benchmark, SIRBench-V1, to evaluate the inductive reasoning abilities of LLMs in scientific settings. Our experimental results show that current LLMs still struggle with this task, underscoring its difficulty and the need for further advancement in this area.[1]

1 Introduction

In recent years, many advanced reasoning models, including OpenAI o1 (OpenAI et al., 2024) and DeepSeek-R1 (DeepSeek-AI et al., 2025), have demonstrated strong *deductive reasoning* capabilities, especially as evidenced by their performance in mathematics and programming tasks. These tasks are typically characterized by concise problem descriptions, where the model is required to generate a long chain of thought (Wei et al., 2022) to solve complex problems.

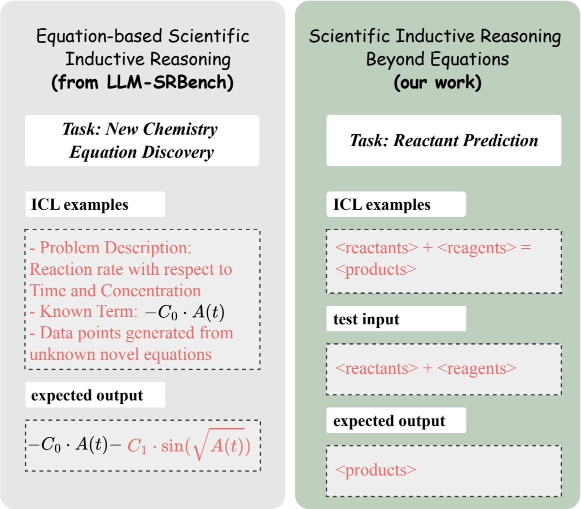
Equation-based Scientific Inductive Reasoning (from LLM-SRBench)
Task: New Chemistry Equation Discovery
ICL examples
- Problem Description: Reaction rate with respect to Time and Concentration
- Known Term: $-C_0 \cdot A(t)$
- Data points generated from unknown novel equations
expected output
$-C_0 \cdot A(t) - C_1 \cdot \sin(\sqrt{A(t)})$
Scientific Inductive Reasoning Beyond Equations (our work)
Task: Reactant Prediction
ICL examples
<reactants> + <reagents> = <products>
test input
<reactants> + <reagents>
expected output
<products>

Figure 1: Illustrative comparison of scientific inductive reasoning: on the left, tasks focused on equation discovery (Shojaee et al., 2025), and on the right, tasks representing broader forms of scientific induction beyond equation generation.

In contrast, *inductive reasoning* (Hayes et al., 2010) poses a different challenge, requiring models to infer general rules or structures from multiple specific observations (Chollet, 2019; Yang et al., 2022). Inductive reasoning involves making predictions about new scenarios based on existing knowledge or observed data (Hayes et al., 2010). Inductive reasoning has been progressively recognized as a critical component for human-like cognitive modeling and the development of general artificial intelligence (Li et al., 2024). However, current LLMs still exhibit notable shortcomings in inductive reasoning tasks (Li et al., 2024; Hua et al., 2025; Yan et al., 2025). Even state-of-the-art models often fail to correctly infer abstract rules from observations and typically rely on memorizing rather than truly understanding the underlying concepts.

Currently, artificial intelligence is increasingly regarded as a transformative paradigm in scientific discovery, with growing applications across disciplines such as physics, materials science, and

[*] Equal contribution.
[†] Corresponding authors.
[1] The open-source code and data can be found at https://github.com/thunlp/SIR-Bench.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 9383–9406
November 4-9, 2025 ©2025 Association for Computational Linguistics

Benchmark	**Task Type**	**Related to Scientific Discovery**	**Beyond Mathematical Equations**	**Closed-Ended Questions**	**#Instances**	**Sequence Length**
MATDESIGN	HI	✓	✓	×	50	250-1,000
TOMATO-Chem	HI	✓	✓	×	51	100-600
ResearchBench	HI	✓	✓	×	1,386	unknown
chaotic systems	SR	✓	×	✓	131	~100
SRSD	SR	✓	×	✓	240	100-300
LLM-SRBench	SR	✓	×	✓	239	~100
MIRAGE	IR	×	✓	✓	2,000	20-100
MIR-Bench	IR	×	✓	✓	6,930	50-250
IOLBench	IR	×	✓	✓	1,500	200-2,000
SIRBench-V1 (Ours)	IR	✓	✓	✓	710	500-3,000

Table 1: Analysis of existing related benchmarks. **HI**: *Hypothetical Induction*, **SR**: *Symbolic Regression*, **IR**: *Inductive Reasoning*. **Related to Scientific Discovery**: targets scientific problem-solving. **Beyond Mathematical Equations**: focuses on reasoning not reducible to equation fitting. **Closed-Ended Questions**: has deterministic answers for automatic evaluation. **#Instances**: number of test examples. **Sequence Length**: input sequence length—crucial as scientific inductive reasoning often requires extracting information from extensive resources.

chemistry (Xu et al., 2021). Against this backdrop, increasing attention has been paid to the inductive reasoning abilities of LLMs in scientific contexts recently (Yang et al., 2024; Liu et al., 2025; Fang et al., 2025). However, systematically leveraging reasoning models to enhance inductive tasks for scientific discovery remains largely underexplored.

While some scientific rules, such as the velocity formula of free fall, can be expressed mathematically, others, such as molecular structure-function relationships, are not readily amenable to such formulation. Under this criterion, we observe that existing LLM-based inductive reasoning research can be broadly categorized based on *whether the underlying rules can be formulated mathematically*. The first category comprises tasks that are mathematical equation-based, which are closely related to symbolic regression (Matsubara et al., 2022; Gilpin, 2021). Recent work has shown that LLMs can serve as equation generators or guide the equation discovery process (Wang et al., 2024; Du et al., 2024; Shojaee et al., 2024, 2025; Fang et al., 2025). However, these tasks typically only cover cases where the underlying rules can be explicitly formulated as equations. A separate line of work targets tasks beyond mathematical equations, proposing new inductive tasks and datasets from various perspectives (Hua et al., 2025; Tang et al., 2024; Banatt et al., 2024; Goyal and Dan, 2025). However, many of these studies emphasize the creation of novel synthetic or low-frequency symbolic systems, which often have a limited connection to discovering scientific patterns in real-world scenarios. Recent efforts under the AI4Science agenda are exploring more scientifically grounded settings where models emulate researchers by deriving insights or hypotheses from scientific materials (Yang et al., 2023, 2024; Liu et al., 2025). However, the reasoning processes of these studies often remain coarse-grained or open-ended, making robust automatic evaluation challenging.

To address these gaps, we propose to examine the capabilities of LLMs in *Scientific Inductive Reasoning Tasks Beyond Mathematical Equations*. To the best of our knowledge, high-quality and easy-to-evaluate datasets to directly investigate this problem are currently lacking. We have therefore created *SIRBench-V1*, a new benchmark consisting of a series of subtasks in chemistry and biology. In these subtasks, the underlying rules cannot be expressed through mathematical equations, yet they yield relatively deterministic answers. We transform basic scientific resources from prior studies (Grešová et al., 2023; Liu et al., 2024; Guo et al., 2023; Edwards et al., 2022a; Irwin et al., 2021; Westerlund et al., 2024b,a; Kim et al., 2018) into inductive reasoning tasks. Furthermore, to eliminate LLM memorization, we design counterfactual tasks that establish synthetic scientific rules for the models to reason with, rather than recall.

We follow several commonly adopted reasoning strategies for LLMs on the SIRBench-V1, including implicit and explicit reasoning, self-consistency (Wang et al., 2022), and hypothesis refinement (Qiu et al., 2023). By investigating the performance of several LLMs augmented with different reasoning strategies, we find that equation-free scientific inductive reasoning is highly challenging for modern LLMs. Gemini-2.5-Flash, the best-performing model, achieves an average accuracy of 43.81% in our benchmark, while Claude-3.5-Haiku and GPT-4.1 demonstrate a lower average accuracy of 31.53% and 32.41%, respectively. We also observe that using sophisticated reasoning strategies provides minimal performance improvement and, in some cases, even leads to performance decline. Using hypothesis refinement, Gemini-2.5-Flash, Claude-3.5-Haiku, and GPT-4.1 attain an average accuracy of 39.06%, 31.63%, and 33.25%, respectively. We believe this work will pave the way for a new and fruitful avenue of research in scientific discovery.

Contributions In summary, the main contributions of this work are as follows:

- We present SIRBench-V1, a new scientific inductive reasoning benchmark featuring authentic and counterfactual test examples from tasks in both biology and chemistry.
- We conduct evaluations using several representative LLMs in conjunction with diverse advanced inference strategies, the results of which demonstrate the capability boundaries of the examined LLMs.
- We derive several constructive findings for scientific inductive reasoning, such as a comparison between many-short-shot and long-few-shot learning approaches and an analysis of memorization, which we anticipate will be helpful for subsequent studies.

2 Related Work

2.1 Inductive Reasoning

Benchmark Various benchmarks have recently been introduced to systematically evaluate these capabilities from multiple perspectives. Hua et al. (2025) evaluate the model's ability to infer string transformation rules from limited input-output examples. Bongard-OpenWorld (Wu et al., 2023) examines conceptual induction and image classification in few-shot scenarios. Tang et al. (2024) propose an embodied interactive environment requiring models to induce task rules and objectives. MIR-Bench (Yan et al., 2025) provides a many-shot in-context benchmark covering various function-based input-output pairs. WILT (Banatt et al., 2024), inspired by the Wason 2-4-6 task, evaluates multi-turn inductive reasoning and generalization capabilities. Additionally, benchmarks such as LINGOLY (Bean et al., 2024), Linguini (Sánchez et al., 2024) and IOLBench (Goyal and Dan, 2025), derived from the International Linguistics Olympiad, challenge model generalization under low-resource language scenarios.

Methods Beyond benchmark development, recent efforts have also explored structured frameworks to enhance inductive reasoning in LLMs, addressing limitations observed with chain-of-thought prompting and few-shot methods (Bowen et al., 2024; Gendron et al., 2023). For instance, Chain-of-Language-Models (Yang et al., 2022) employs a modular pipeline integrating rule generation and verification. Qiu et al. (2023) combines LLMs with symbolic executors in a propose-verify-refine loop, significantly enhancing robustness. Similarly, the De-In-Ductive (DID) (Cai et al., 2024) simulates a human-like inductive-then-deductive reasoning sequence within a single prompt, enabling flexible strategy switching and improved cross-task generalization.

2.2 Scientific Inductive Reasoning in LLMs

Symbolic Regression Symbolic regression is a core approach for scientific discovery (Matsubara et al., 2022; Gilpin, 2021). It is valued for its ability to extract analytical expressions directly from data (Angelis et al., 2023). Recent studies have extended this paradigm by incorporating LLMs into the tasks. In materials science, Wang et al. (2024) highlight its role in revealing underlying physical and chemical principles. Du et al. (2024) propose a prompt-based framework using LLMs to generate candidate equations, offering greater flexibility than traditional methods. Shojaee et al. (2024) treat equations as programs, guided by scientific priors. To support systematic evaluation, they then introduce LLM-SRBench, a multi-domain benchmark designed to evaluate LLMs' true discovery capabilities.

Hypothetical Induction Hypothetical Induction has been recognized as a subtask of inductive reasoning (Norton, 2003), with growing interest in using LLMs to generate novel, valuable scientific hypotheses from background knowledge or observations. Kumbhar et al. (2025) introduced a goal-driven dataset and evaluation framework in materials science, while Yang et al. (2023, 2024) constructed datasets for hypothesis generation in chemistry and social science. Researchbench (Liu et al., 2025) further provides the first benchmark covering inspiration retrieval, hypothesis formulation, and ranking.

3 SIRBench-V1: Task and Construction

We curate 7 tasks, with 100 samples for each biology task, including synthetic tasks, and 30 samples for each chemistry task.

3.1 Task Overview

Task 1: DNA Translation (Synthetic) This task simulates the biological process of translating a DNA sequence into its corresponding amino acid sequence. The model is required to induce the codon-to-amino-acid mappings solely based on in-context learning (ICL) examples and apply the inferred mappings to translate a target DNA sequence. However, LLMs may have internalized the canonical genetic codon table as prior knowledge, enabling them to generate the correct amino acid sequence through memorization rather than genuine rule induction. To better assess the inductive reasoning capabilities of the model, we provide a synthetic alternative to the standard task design, by randomly assigning codon-to-amino-acid mappings.

Task 2: DNA Table Inference (Synthetic) This task focuses explicitly on evaluating the model's inductive ability by requiring it to recover the underlying codon table based solely on a set of DNA–amino acid sequence pairs. The model is asked to infer the translation rules and provide a fully structured codon table, including codon-to-amino acid mappings, start codons, and stop codons. We follow the same design as in Task 1, providing both standard and synthetic configurations.

Task 3: DNA Transformation This task adopts a fully synthetic setup, with the goal of evaluating the model's ability to infer transformation rules from ICL examples and to apply them correctly to unseen test sequences. Each ICL example consists of an input–output DNA sequence pair generated by applying one of several predefined transformations: sequence reversal, complementation, reverse complementation, segmented transformation, and fixed base mutation.

Task 4: Molecule Design This task requires LLMs to generate molecular structures that satisfy a given textual description. The input is a natural language sentence (in English), and the output is the corresponding molecule represented in SMILES format.

Task 5: Molecule Captioning This task is the inverse of Task 4, where the input is a molecular structure and the model is expected to generate a corresponding description or annotation in natural language.

Task 6: Reaction Prediction This task focuses on chemical reaction prediction. Given one or more reactants and reagents, the model is expected to predict the resulting product in the form of a SMILES string.

Task 7: Name Prediction This task focuses on conversions between three common chemical representations: SMILES (linear structural encodings), IUPAC names (standardized nomenclature), and molecular formulas (atomic composition). We include four relatively unambiguous conversions: *smiles2formula*, *smiles2iupac*, *iupac2smiles*, and *iupac2formula*.

3.2 Data Collection

Biology We derive source DNA sequences and their corresponding amino acid sequences from GenomicLLM_GRCh38 (Grešová et al., 2023; Liu et al., 2024) for the standard task. For the synthetic task, we generate codon tables by randomizing every mapping except the start and stop codons, and translate inputs using these tables.

For DNA Transformation, we randomly sample DNA fragments from the training set as ICL examples and truncate them to a maximum length, and do the same for test sequences. The transformation type and base-pairing schemes are randomly sampled from a predefined set. These base-pairing schemes are designed manually to disrupt natural complementarity, increasing the inductive reasoning challenge. For all the tasks, we ensure that the ICL examples cover all the mappings used in the test example.

Chemistry ChemLLMBench (Guo et al., 2023) is a chemistry-domian LLM benchmark comprising eight tasks. We select four tasks, corresponding to Task 4-7 in our work, which exhibit a relatively stronger emphasis on inductive reasoning capabilities. The Molecule Design and Captioning tasks are based on the ChEBI-20 dataset (Edwards et al., 2022a), pairing molecular SMILES with textual description. The Reaction Prediction task draws

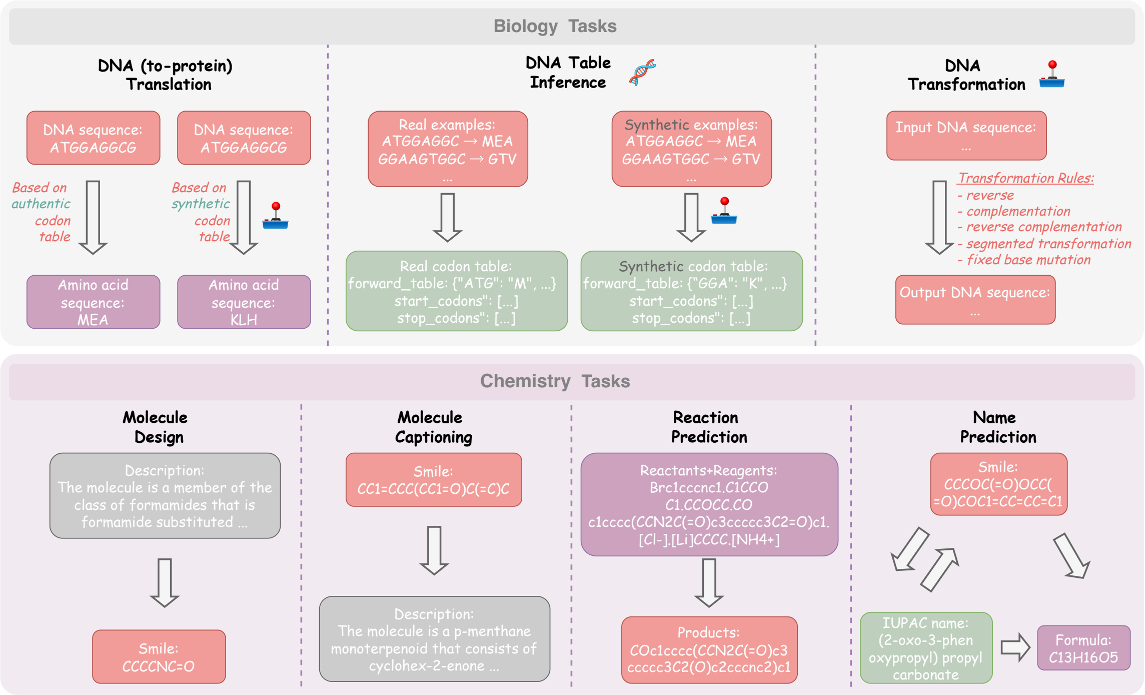
Biology Tasks
DNA (to-protein) Translation
DNA sequence: ATGGAGGCG
DNA sequence: ATGGAGGCG
Based on authentic codon table
Based on synthetic codon table
Amino acid sequence: MEA
Amino acid sequence: KLH
DNA Table Inference
Real examples: ATGGAGGC → MEA GGAAGTGGC → GTV ...
Synthetic examples: ATGGAGGC → MEA GGAAGTGGC → GTV ...
Real codon table: forward_table: {"ATG": "M", ...} start_codons": [...] stop_codons": [...]
Synthetic codon table: forward_table: {"GGA": "K", ...} start_codons": [...] stop_codons": [...]
DNA Transformation
Input DNA sequence: ...
Transformation Rules:
- reverse
- complementation
- reverse complementation
- segmented transformation
- fixed base mutation
Output DNA sequence: ...
Chemistry Tasks
Molecule Design
Description: The molecule is a member of the class of formamides that is formamide substituted ...
Smile: CCCCNC=O
Molecule Captioning
Smile: CC1=CCC(CC1=O)C(=C)C
Description: The molecule is a p-menthane monoterpenoid that consists of cyclohex-2-enone ...
Reaction Prediction
Reactants+Reagents: Brc1cccnc1.C1CCO C1.CCOCC.CO c1cccc(CCN2C(=O)c3ccccc3C2=O)c1. [Cl-].[Li]CCCC.[NH4+]
Products: COc1cccc(CCN2C(=O)c3 ccccc3C2(O)c2cccnc2)c1
Name Prediction
Smile: CCCOC(=O)OCC(=O)COC1=CC=CC=C1
IUPAC name: (2-oxo-3-phen oxypropyl) propyl carbonate
Formula: C13H16O5

Figure 2: Our benchmark includes 7 tasks spanning two scientific disciplines: biology and chemistry. 🕹️ denotes tasks that adopt a synthetic configuration; 🧬 refers to tasks that involve only rule induction from examples, while others involve both induction and application to a new test input.

on the USPTO-MIT Mixed reaction dataset (Irwin et al., 2021; Westerlund et al., 2024b,a), which contains information on reactants, reagents, and products in SMILES reaction format. The Name Prediction task is derived from PubChem (Kim et al., 2018), which offers extensive mappings between SMILES strings and their corresponding standard chemical names, including both IUPAC names and molecular formulas.

3.3 Metrics

Biology All three tasks are evaluated using accuracy as the primary metric, computed as the proportion of correctly predictions.

Chemistry For molecule design, we adopt eight metrics, including BLEU, Exact Match (Edwards et al., 2022b), and Levenshtein distance (Miller et al., 2009) for string-level consistency; validity for structural correctness; MACCS (Ratcliff and Metzener, 1988), RDK (Landrum, 2020), and Morgan (Dash et al., 2023) for structural similarity; and FCD (Preuer et al., 2018) for distributional similarity. For molecule captioning, we use BLEU, ROUGE, and METEOR to capture surface-level overlaps, but also introduce an LLM-as-a-Judge score (1–10 scale), with an emphasis on scientific accuracy, while also considering completeness and clarity. For reaction prediction, we follow the Top-1 Accuracy metric and improve robustness by canonicalizing both predicted and reference SMILES using RDKit (Landrum, 2020) before comparison. Finally, for name prediction, we apply the same canonicalization for the *iupac2smiles* task, and adopt Exact Match Accuracy for the other three tasks (*smiles2formula*, *smiles2iupac*, and *iupac2formula*).

4 Evaluation

4.1 Models

In order to provide a comprehensive assessment of the inductive reasoning capabilities of cost-optimized, flagship, and reasoning LLMs, we choose one representative model from each category, namely Claude-3.5-Haiku, GPT-4.1, and Gemini-2.5-Flash. Since our benchmark is integrated into the OpenCompass framework, it can be easily evaluated on any other LLM. To ensure consistency and encourage output diversity during repeated sampling, we set the temperature at 1.0 for all experiments. For Gemini-2.5-Flash, we retain its default “thinking” configuration.

4.2 Inference Strategies

We evaluate SIRBench-V1 on four commonly used inference strategies for inductive reasoning as illustrated in figure 3. Explicit inductive reasoning serves as a baseline for advanced methods like self-consistency and hypothesis refinement, where the LLM needs to explicitly formulate and apply the hypotheses.

Implicit Inductive Reasoning. We provide the LLM with ICL examples and ask the LLM to provide the final answer directly without explicitly stating the induced rules. This approach is the most straightforward way to perform inductive reasoning.

Explicit Inductive Reasoning. We prompt the LLM to formulate a hypothesis based on the ICL examples. Then, we let the LLM apply the hypothesis to the given target question to obtain the final answer. This approach forces the LLM to perform the inductive reasoning process explicitly.

Self-Consistency. For self-consistency (Wang et al., 2022), we sample multiple hypotheses (we use $n = 5$) from the LLM and ask it to apply each of them to the target question, obtaining a corresponding answer from each hypothesis. A final answer is selected using majority voting performed by the LLM itself via prompting (see appendix C).

Hypothesis Refinement. The hypothesis refinement method (Qiu et al., 2023) follows a three-stage iterative process: hypothesis generation, selection, and refinement.

Initially, we sample multiple hypotheses ($n = 5$) based on the ICL examples, then evaluate them using one of the two approaches: (1) for code-executable tasks, we translate them into Python functions and execute them following Qiu et al. (2023), or (2) otherwise, we have the LLM apply each hypothesis directly. A task-specific evaluator scores each hypothesis's output.

Next, we generate a new set of hypotheses ($n = 5$) by prompting (see appendix C for prompt) the LLM to refine the highest-scoring hypothesis based on feedback.

We repeat this select-and-refine loop up to $t = 3$ iterations, stopping early if the hypothesis achieves a perfect score on ICL examples or performance degradation is detected. We added the early stopping mechanism for performance degradation to prevent weaker models from degrading rule quality.

Finally, we apply the best resulting hypothesis to the target question to produce the answer.

5 Results and Analysis

5.1 Main Results

Table 2 reveals consistently low performance across most tasks, highlighting the limitations of current LLMs in scientific inductive reasoning tasks beyond mathematical equations. Among the evaluated models, Gemini-2.5-Flash demonstrates superior performance in computationally intensive tasks while exhibiting comparable results to other models in conceptually oriented tasks such as Molecule Caption. Additionally, larger flagship models perform better than cost-optimized models.

We observe that LLMs struggle with explicit inductive reasoning (i.e., proposing effective rules and applying them to novel inputs), as shown by the performance drop from implicit to explicit inductive reasoning. Self-consistency helps alleviate this shortcoming by sampling multiple diverse reasoning paths and marginalizing across them, thereby enhancing the robustness of the explicit inductive reasoning process. The hypothesis refinement strategy further improves the performance, as it selects the best rule from multiple sampled hypothesis and revises the rule at each iteration. However, we find that the advantage of hypothesis refinement over implicit inductive reasoning varies inconsistently across tasks and models.

To validate our findings across more LLMs, we evaluated additional open-source models under implicit inductive reasoning, as shown in Table 3. Deepseek-V3-0324 performs comparably with GPT-4.1 across most tasks, while Qwen3-8B with thinking generates extremely long chain-of-thought reasoning for biology tasks, often exceeding its recommended 32K max output length without completing the reasoning process, demonstrating that long chain-of-thought is not effective on the biology tasks. These results reinforce our findings on the fundamental limitation of current LLMs in scientific inductive reasoning. Additionally, current inductive reasoning methods remain inadequate for scientific inductive reasoning tasks beyond mathematical equations.

5.2 Effect of Length

Being able to perform inductive reasoning on a long context is fundamental. We evaluated the LLMs on DNA transformation and DNA translation tasks

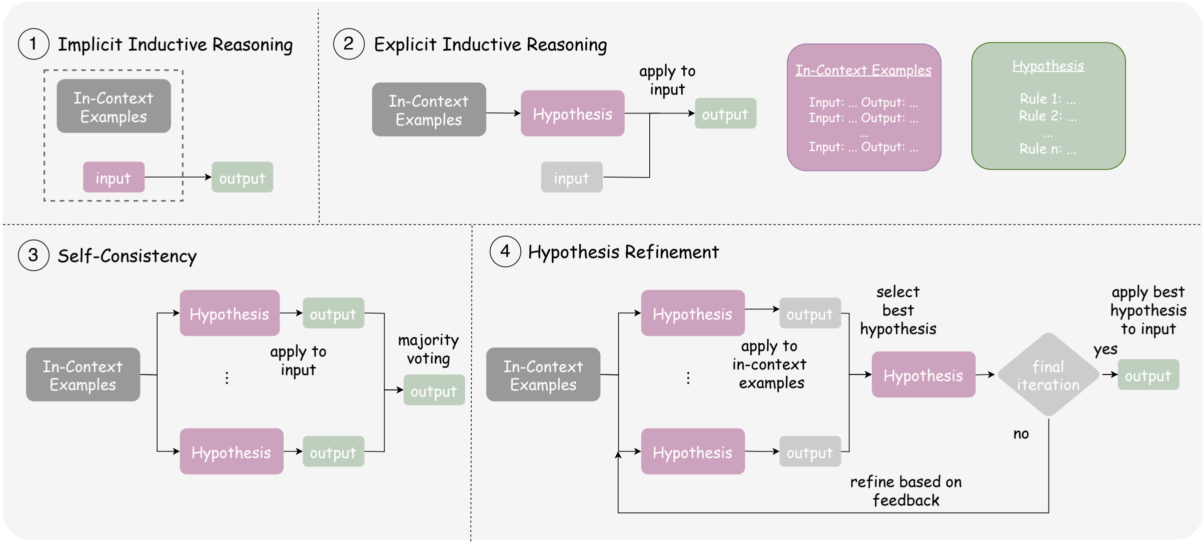

Figure 3: Comparison of four inference strategies: (1) Implicit induction - directly providing output; (2) Explicit induction - formulating clear hypotheses explicitly; (3) Self-consistency - using multiple reasoning paths to reach consensus; and (4) Hypothesis refinement - iteratively improving hypothesis on feedback.

Models	Biology			Chemistry				Avg.
	DNA Translation	DNA Table Inference	DNA Transformation	Molecule Design	Molecule Caption	Reaction Prediction	Name Prediction	
Implicit Inductive Reasoning								
Claude-3.5-Haiku	5.47	10.23	27.28	62.00	67.70	44.44	3.57	31.53
GPT-4.1	5.71	12.73	31.37	75.00	66.30	22.22	13.51	32.41
Gemini-2.5-Flash	**11.72**	**32.06**	30.42	**85.00**	63.30	**54.17**	30.00	**43.81**
Explicit Inductive Reasoning								
Claude-3.5-Haiku	5.85	9.72	26.05	64.00	54.00	19.23	2.81	25.95
GPT-4.1	5.31	12.13	28.73	69.00	59.00	17.86	6.09	28.30
Gemini-2.5-Flash	9.14	23.34	28.66	77.00	67.70	34.78	30.00	38.66
Self-Consistency (Wang et al., 2022)								
Claude-3.5-Haiku	5.11	10.00	26.34	66.00	69.70	20.83	0.83	28.40
GPT-4.1	5.96	13.19	30.81	72.00	65.70	25.00	9.58	31.75
Gemini-2.5-Flash	9.15	24.84	30.4	80.00	**70.00**	39.29	**40.13**	41.97
Hypothesis Refinement (Qiu et al., 2023)								
Claude-3.5-Haiku	5.79	10.02	30.05	73.00	**72.70**	28.00	1.88	31.63
GPT-4.1	5.62	14.57	**35.56**	67.00	66.30	32.14	11.59	33.25
Gemini-2.5-Flash	10.60	28.55	30.37	72.00	65.70	32.14	34.07	39.06

Table 2: Performance of Claude-3.5-Haiku, GPT-4.1, and Gemini-2.5-Flash on SIRBench-V1 using four inference strategies. All scores report accuracy (%), except Molecule Design (Morgan similarity rescaled to 0-100). Molecule Caption reports the accuracy from LLM-as-judge. Synthetic versions were used for DNA Translation and DNA Table Inference tasks.

with varying sequence length configurations. The DNA transformation task demands the comprehension of the entire sequence (e.g., identifying reversals), while the DNA translation task requires observation of local patterns. As shown in figure 4, for DNA transformation, we found that the LLMs achieve relatively strong performance on shorter sequences but exhibits a significant performance decline as sequence length increases. For DNA translation, GPT-4.1 and Claude-3.5-Haiku show minimal decrease with longer sequences only because they struggle with this task at shorter lengths. The results indicate that current LLMs are effective at inducing pattern only within limited input

Models	Biology			Chemistry				Avg.
	DNA Translation	DNA Table Inference	DNA Transformation	Molecule Design	Molecule Caption	Reaction Prediction	Name Prediction	
Qwen3-8B (with thinking)	0.20	4.88	3.24	59.00	52.67	3.33	1.67	17.00
Qwen3-8B (without thinking)	6.30	7.06	27.19	50.00	49.67	0.00	0.00	20.03
Deepseek-V3-0324	7.21	12.24	28.81	75.00	64.00	30.00	14.17	33.06

Table 3: Performance of Qwen3-8B and Deepseek-V3-0324 on SIRBench-V1 under the **Implicit Inductive Reasoning** setting. Scores are accuracy (%) except Molecule Design (Morgan similarity, 0-100 scale) and Molecule Caption (LLM-as-judge accuracy). Synthetic versions used for DNA tasks.

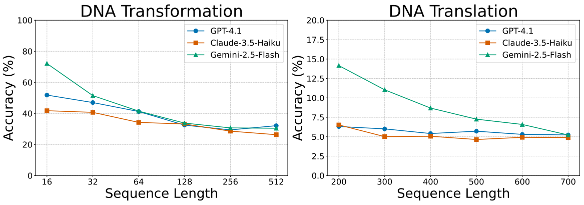

Figure 4: Effect of Sequence Length in Transformation and DNA Translation tasks.

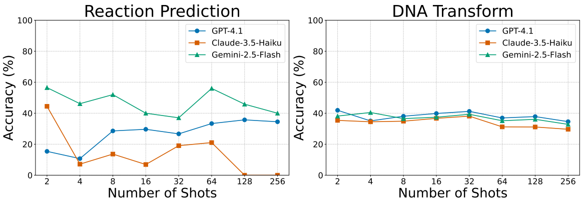

Figure 5: Effect of Number of Shots in Reaction Prediction and DNA Transformation tasks.

lengths. This limitation reflects the broader challenge of developing robust inductive reasoning capabilities that can handle long context.

5.3 Effect of Number of Shots

We examine the effect of the number of shots on accuracy in one representative task each from the domains of biology and chemistry. Figure 5 shows that increasing the number of shots has varying effects on different models. In reaction prediction task, GPT-4.1 exhibits an upward trend, showing that it benefits from additional shots. In contrast, Claude-3.5-Haiku shows performance degradation, likely due to limitations in its context processing capability. Gemini-2.5-Flash does not show any clear upward or downward trend with as shot increases. For DNA transformation, all the models exhibit consistent performance, implying that additional examples provide limited benefit.

5.4 Many-Short-Shot vs. Long-Few-Shot

Unlike previous studies that only explore increasing the number of relatively short examples (Yan et al., 2025), we also explore the inductive reasoning capabilities of LLMs on few long examples. The latter paradigm adheres more to real-world applications, where it is difficult to obtain numerous examples for long input tasks. Our comparative analysis in table 4 across both scenarios while maintaining the total input length demonstrates that LLMs perform worse with few long examples. This finding highlights a critical area for the advancement of LLM inductive reasoning ability.

Model	Many-Short-Shot	Few-Long-Short
Claude-3.5-Haiku	31.19	15.63
GPT-4.1	36.94	25.64
Gemini-2.5-Flash	35.14	24.47

Table 4: Performance comparison in many-short-shot versus long-few-shot settings on the DNA Translation task. The many-short-shot setting uses 64 shots with sequence length 100, while the few-long-shot setting uses 4 shots with sequence length 1600.

5.5 Task Difficulty Analysis

Reasoning ability is not only reflected in overall accuracy but also in performance across difficulty levels. We analyzed two representative tasks, one from biology and one from chemistry, under Implicit Inductive Reasoning. Test instances were categorized into Easy, Medium, Hard, with 100 samples each. The DNA Translation samples were grouped by input sequence length, with ranges of 100-300 for Easy, 300-500 for Medium, and 500-700 for Hard, while the Molecule Design samples were classified by molecular complexity using RDKit based on structural features. As shown in both Table 5 and Table 6, model performance exhibits a clear downward trend from easy to hard samples, suggesting that difficulty-based categorization offers a straightforward way to assess robustness while also enabling a more fine-grained evaluation of reasoning abilities across domains.

Difficulty Level		GPT-4.1	Claude	Gemini
Easy	accuracy	6.16	5.77	12.6
Medium	accuracy	5.56	4.85	7.98
Hard	accuracy	5.27	4.91	5.9

Table 5: Performance of LLMs on the DNA Translation task by difficulty level.

Difficulty Level		GPT-4.1	Claude	Gemini
Easy	validity	0.94	0.67	0.94
	morgan_sims	0.67	0.39	0.89
	fcd (↓)	2.66	9.82	1.15
Medium	validity	0.92	0.64	0.88
	morgan_sims	0.55	0.29	0.78
	fcd (↓)	7.77	21.08	4.73
Hard	validity	0.74	0.59	0.41
	morgan_sims	0.46	0.21	0.6
	fcd (↓)	19.85	29.86	22.24

Table 6: Performance of LLMs on the Molecule Design task by difficulty level.

5.6 Counterfactual Evaluation

Model	DNA Translation		DNA Table Inf.	
	Aut.	Syn. (Δ)	Aut.	Syn. (Δ)
Claude-3.5-Haiku	21.95	5.47 (-16.48)	68.50	10.23 (-58.27)
GPT-4.1	21.24	5.71 (-15.53)	81.84	12.73 (-69.11)
Gemini-2.5-Flash	30.64	11.72 (-18.92)	87.09	32.06 (-55.03)

Table 7: Performance comparison between authentic and synthetic versions of chosen tasks. Δ represents the performance gap, calculated as the score on synthetic tasks minus the score on authentic tasks.

To investigate whether LLMs perform true inductive reasoning, we compare their performance on original and synthetic settings of DNA Translation and Table Inference. As illustrated in Table 7, all three models suffer a dramatic performance decline in synthetic tasks, suggesting that higher performance in authentic versions stems from the memorization of standard mappings rather than genuine inductive reasoning capabilities.

Among the evaluated models, Gemini-2.5-Flash maintains the highest performance on both original and synthetic versions of the tasks. This suggests that reasoning models have better capability to identify rules beyond the constraints of memorized knowledge than non-reasoning models. However, its absolute score in synthetic tasks remains low. Overall, these results indicate that current LLMs are fundamentally limited in their ability to perform genuine inductive reasoning. In the context of scientific discovery, LLMs need to recognize novel patterns rather than just retrieve existing knowledge. Therefore, our findings highlight the need to distinguish inductive reasoning from retrieval to advance the ability of LLMs for scientific discovery.

6 Conclusion

In this paper, we introduce SIRBench-V1, a benchmark that includes Chemistry and Biology subtasks, to evaluate the scientific inductive reasoning of LLMs on tasks beyond mathematical equation. We evaluated different LLMs using commonly used reasoning strategies on our proposed benchmark. We found that current leading LLMs obtain low performance on our benchmark and that using sophisticated strategies provide minimal benefits. Additionally, we point out limitations of LLMs in performing inductive reasoning on longer context lengths, few-long-shot settings, and counterfactual rules. The experimental results will provide valuable insights for future studies on LLM-driven scientific discovery.

7 Limitations

In this work, we take the first step toward incorporating scientific scenarios into the design of the LLM-Based Inductive Reasoning Beyond Equations and introduce a new dataset for evaluation. However, the SIRBench-V1 is limited to chemistry and biology domains. As a next step, we plan to invite domain experts in these areas to review and refine both our benchmark and evaluation protocol. In the future, we aim to expand the benchmark to cover a broader range of scientific disciplines.

Acknowledgement

This work is supported by the AI9Stars community, the Fundamental Research Funds for the Central Universities, and the Research Funds of Beijing Language and Culture University (25YCX118). We also thank the anonymous reviewers in ACL Rolling Review May 2025. Their insightful feedback and suggestions are significant to refining and improving our work.

References

Dimitrios Angelis, Filippos Sofos, and Theodoros E. Karakasidis. 2023. Artificial intelligence in physical sciences: Symbolic regression trends and perspectives. *Archives of Computational Methods in Engineering*, pages 1 – 21.

Eryk Banatt, Jonathan Cheng, Skanda Vaidyanath, and Tiffany Hwu. 2024. Wilt: A multi-turn, memorization-robust inductive logic benchmark for llms. *ArXiv*, abs/2410.10998.

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

Chen Bowen, Rune Sætre, and Yusuke Miyao. 2024. A comprehensive evaluation of inductive reasoning capabilities and problem solving in large language models. In *Findings*.

Chengkun Cai, Xu Zhao, Haoliang Liu, Zhongyu Jiang, Tianfang Zhang, Zongkai Wu, Jenq-Neng Hwang, and Lei Li. 2024. The role of deductive and inductive reasoning in large language models. *ArXiv*, abs/2410.02892.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *ArXiv*, abs/2311.17311.

François Chollet. 2019. On the measure of intelligence. *Preprint*, arXiv:1911.01547.

Debadutta Dash, Rahul Thapa, J. Banda, Akshay Swaminathan, Morgan Cheatham, Mehr Kashyap, Nikesh Kotecha, Jonathan H. Chen, Saurabh Gombar, Lance Downing, Rachel A. Pedreira, Ethan Goh, Angel Arnaout, Garret K. Morris, H Magon, Matthew P. Lungren, Eric Horvitz, and Nigam H. Shah. 2023. Evaluation of gpt-3.5 and gpt-4 for supporting real-world information needs in healthcare delivery. *ArXiv*, abs/2304.13714.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

Mengge Du, Yuntian Chen, Zhongzheng Wang, Longfeng Nie, and Dong juan Zhang. 2024. Large language models for automatic equation discovery of nonlinear dynamics. *Physics of Fluids*.

Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. 2022a. Translation between molecules and natural language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 375–413, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Carl N. Edwards, T. Lai, Kevin Ros, Garrett Honke, and Heng Ji. 2022b. Translation between molecules and natural language. *ArXiv*, abs/2204.11817.

You-Le Fang, Dong-Shan Jian, Xiang Li, and Yan-Qing Ma. 2025. Ai-newton: A concept-driven physical law discovery system without prior physical knowledge. *Preprint*, arXiv:2504.01538.

Gaël Gendron, Qiming Bao, M. Witbrock, and Gillian Dobbie. 2023. Large language models are not strong abstract reasoners. In *International Joint Conference on Artificial Intelligence*.

William Gilpin. 2021. Chaos as an interpretable benchmark for forecasting and data-driven modelling. *ArXiv*, abs/2110.05266.

Satyam Goyal and Soham Dan. 2025. Iolbench: Benchmarking llms on linguistic reasoning. *ArXiv*, abs/2501.04249.

Katarína Grešová, Vlastimil Martinek, David Čechák, Petr Šimeček, and Panagiotis Alexiou. 2023. Genomic benchmarks: a collection of datasets for genomic sequence classification. *BMC Genomic Data*, 24(1):25.

Taicheng Guo, Kehan Guo, Bozhao Nan, Zhengwen Liang, Zhichun Guo, N. Chawla, O. Wiest, and Xiangliang Zhang. 2023. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. In *Neural Information Processing Systems*.

Brett K. Hayes, Evan Heit, and Haruka Swendsen. 2010. Inductive reasoning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(2):278–292.

Wenyue Hua, Tyler Wong, Sun Fei, Liangming Pan, Adam Jardine, and William Yang Wang. 2025. Inductionbench: Llms fail in the simplest complexity class. *ArXiv*, abs/2502.15823.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. 2021. Chemformer: a pretrained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, Paul A. Thiessen, Bo Yu, Leonid Y. Zaslavsky, Jian Zhang, and Evan E. Bolton. 2018. Pubchem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47:D1102 – D1109.

Shrinidhi Kumbhar, Venkatesh Mishra, Kevin Coutinho, Divij Handa, Ashif Iquebal, and Chitta Baral. 2025. Hypothesis generation for materials discovery and design using goal-driven and constraint-guided llm agents. *ArXiv*, abs/2501.13299.

Greg Landrum. 2020. Rdkit: Open-source cheminformatics. http://www.rdkit.org. [Online; accessed 14-May-2025].

Jiachun Li, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Mirage: Evaluating and explaining inductive reasoning process in language models. *ArXiv*, abs/2410.09542.

Huaqing Liu, Shuxian Zhou, Peiyi Chen, Jiahui Liu, Ku-Geng Huo, and Lanqing Han. 2024. Exploring genomic large language models: Bridging the gap between natural language and gene sequences. *bioRxiv*.

Yujie Liu, Zonglin Yang, Tong Xie, Jinjie Ni, Ben Gao, Yuqiang Li, Shixiang Tang, Wanli Ouyang, Erik Cambria, and Dongzhan Zhou. 2025. Researchbench: Benchmarking llms in scientific discovery via inspiration-based task decomposition. *ArXiv*, abs/2503.21248.

Yoshitomo Matsubara, Naoya Chiba, Ryo Igarashi, Tatsunori Taniai, and Y. Ushiku. 2022. Rethinking symbolic regression datasets and benchmarks for scientific discovery. *ArXiv*, abs/2206.10540.

Frederic P. Miller, Agnes F. Vandome, and John McBrewster. 2009. Levenshtein distance: Information theory, computer science, string (computer science), string metric, damerau?levenshtein distance, spell checker, hamming distance.

John D. Norton. 2003. A little survey of induction.

OpenAI, :, Aaron Jaech, Adam Kalai, and *et al.*. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. 2018. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 58 9:1736–1741.

Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. 2023. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. *ArXiv*, abs/2310.08559.

John W. Ratcliff and David E. Metzener. 1988. Pattern matching: The gestalt approach. *Dr. Dobb's Journal*, 13(7):46.

Eduardo Sánchez, Belen Alastruey, Christophe Ropers, Pontus Stenetorp, Mikel Artetxe, and Marta Ruiz Costa-jussà. 2024. Linguini: A benchmark for language-agnostic linguistic reasoning. *ArXiv*, abs/2409.12126.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. 2024. Llm-sr: Scientific equation discovery via programming with large language models. *ArXiv*, abs/2404.18400.

Parshin Shojaee, Ngoc-Hieu Nguyen, Kazem Meidani, Amir Barati Farimani, Khoa D. Doan, and Chandan K. Reddy. 2025. Llm-srbench: A new benchmark for scientific equation discovery with large language models.

Xiaojuan Tang, Jiaqi Li, Yitao Liang, Song chun Zhu, Muhan Zhang, and Zilong Zheng. 2024. Mars: Situated inductive reasoning in an open-world environment. *ArXiv*, abs/2410.08126.

Guanjie Wang, Erpeng Wang, Zefeng Li, Jian Zhou, and Zhimei Sun. 2024. Exploring the mathematic equations behind the materials science data using interpretable symbolic regression. *Interdisciplinary Materials*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Andreas M. Westerlund, Lakshman Saigiridharan, and Samuel Genheden. 2024a. Constrained synthesis planning with disconnection-aware transformer and multi-objective search. *ChemRxiv*. Preprint, not peer-reviewed.

Annie M. Westerlund, Siva Manohar Koki, Supriya Kancharla, Alessandro Tibo, Lakshidaa Saigiridharan, Mikhail Kabeshov, Rocío Mercado, and Samuel Genheden. 2024b. Do chemformers dream of organic matter? evaluating a transformer model for multistep retrosynthesis. *Journal of chemical information and modeling*.

Rujie Wu, Xiaojian Ma, Qing Li, Wei Wang, Zhenliang Zhang, Song-Chun Zhu, and Yizhou Wang. 2023. Bongard-openworld: Few-shot reasoning for free-form visual concepts in the real world. *ArXiv*, abs/2310.10207.

Yongjun Xu, Qi Wang, Zhulin An, Fei Wang, Libo Zhang, Yanjun Wu, Fengliang Dong, Cheng-Wei Qiu, Xin Liu, Junjun Qiu, Keqin Hua, Wentao Su, Huiyu Xu, Yong Han, Xinya Cao, En ju Liu, Chenguang Fu, Zhigang Yin, Miao Liu, and 28 others. 2021. Artificial intelligence: A powerful paradigm for scientific research. *The Innovation*, 2.

Kai Yan, Zhan Ling, Kang Liu, Yifan Yang, Ting-Han Fan, Lingfeng Shen, Zhengyin Du, and Jiecao Chen. 2025. Mir-bench: Benchmarking llm's long-context intelligence via many-shot in-context inductive reasoning. *ArXiv*, abs/2502.09933.

Zonglin Yang, Li Dong, Xinya Du, Hao Cheng, E. Cambria, Xiaodong Liu, Jianfeng Gao, and Furu Wei. 2022. Language models as inductive reasoners. *ArXiv*, abs/2212.10923.

Zonglin Yang, Xinya Du, Junxian Li, Jie Zheng, Soujanya Poria, and E. Cambria. 2023. Large language models for automated open-domain scientific hypotheses discovery. In *Annual Meeting of the Association for Computational Linguistics*.

Zonglin Yang, Wanhao Liu, Ben Gao, Tong Xie, Yuqiang Li, Wanli Ouyang, Soujanya Poria, Erik Cambria, and Dongzhan Zhou. 2024. Moosechem: Large language models for rediscovering unseen chemistry scientific hypotheses. *ArXiv*, abs/2410.07076.

A Additional Details on SIRBench-V1

A.1 Dataset Configurations

We curate 7 tasks in total. Considering that multiple metrics provide robust assessment, for chemistry tasks, we evaluate Molecule Design, Molecule Captioning and Reaction Prediction with 30 examples each. For Name Prediction, we sample 30 examples for each type of transformation (including *smiles2formula*, *smiles2iupac*, *iupac2smiles*, and *iupac2formula*). Since biology tasks rely solely on accuracy, we increase the number of examples to 100 for each biology task to ensure more stable evaluation, including DNA Translation, DNA Translation (Synthetic), DNA Table Inference, DNA Table Inference (Synthetic) and DNA Transformation. All experiments are conducted under 5-shot setting, unless otherwise stated. However, since our benchmark has various configurations and supports synthetic data generation for some subtasks, the actual number of items can be configurable.

In our main results, we use the following configurations. For DNA Translation, we uniformly sample across sequence length 200 to 450 since the effective DNA sequences in the dataset starts from length 200. While data are available for longer sequences, only sample until 450 because they are too challenging for most models. For DNA Transformation, we set the sequence length to 300, which is a reasonably challenging level.

A.2 Examples of Transformation Types in DNA Transformation Task

The transformation types include: **1) Sequence reversal**: reversing the order of the entire sequence (e.g., AGCT → TCGA); **2) Complementation**: replacing each base according to a substitution rule (e.g., AGCT → TCGA, using A↔T, C↔G or a randomized complement map); **3) Reverse complementation**: performing complementation followed by reversal (e.g., AGCT → AGCT); **4) Segmented transformation**: transforming fixed-length segments after a fixed stride (e.g., AGCTTAGCGT → AGCTTGACGT, reversing 2 bases every 3 bases); **5) Fixed base mutation**: replacing specific bases with new ones (e.g., AGCT → GGTT, where A→G and C→T).

Task	Model	Initial	Final	Test
DNA Translation	Claude-3.5-Haiku	3.87	6.52	5.79
	GPT-4.1	9.15	11.37	5.62
	Gemini-2.5-Flash	24.37	30.57	10.60
Molecule Design	Claude-3.5-Haiku	0.67	0.71	0.73
	GPT-4.1	0.77	0.82	0.67
	Gemini-2.5-Flash	0.92	0.97	0.72

Table 8: Comparison of initial and final hypothesis quality scores on in-context examples (ICE) alongside corresponding test performance of final hypothesis for various models across DNA Translation (Synth) and Molecule Design tasks. Morgan similarity (scale of 0 to 1) is reported for the Molecule design task.

B Explicit Inductive Reasoning Analysis

B.1 Hypothesis Quality and Refinement

In order to provide a more thorough analysis, we show the computed evaluation score of the generated hypotheses on ICL examples during hypothesis refinement in table 8. For the initial evaluation scores, we report the average score of the best hypothesis generated by the model prior to any refinement. This also serves as an approximate upper bound of the evaluation scores for hypotheses generated by explicit inductive reasoning and self-consistency. We notice that for DNA Translation task, these rules obtained low accuracy on ICL examples. The scores increase at the end of the refinement process, but still remain low. This shows the limitation in inductive reasoning capability of current LLMs. Furthermore, although hypothesis refinement enhances the consistency of rules with ICL examples, we observe that in some cases there remains a substantial gap between performance on ICL examples and generalization to test examples. Future work may focus on enhancing the hypothesis proposal capabilities of LLMs to improve both accuracy on ICL examples and generalization to novel scenarios.

We also provided examples of hypothesis generated by GPT-4.1 which can be found table 9 and 10.

B.2 Misalignment of Advanced Reasoning Strategies

As shown in Table 2, the performance of LLMs does not consistently improve with the application of more fine-grained reasoning strategies. In some cases, advanced strategies even reduce performance. To investigate this phenomenon, we analyzed the recorded reasoning traces, focusing

Task	Hypothesis
DNA Translation	Rule 1: Read the DNA from 5'->3' in consecutive codons (3 bp) starting at the first base. Rule 2: Each codon uniquely maps to one amino acid per the in-context examples. Rule 3: Inferred codon table: ATG:M, ATT:M, ATA:N, ATC:Q, GTT:S, GTC:V, GTA:C, GTG:V, TTT:F, TTC:F, TTA:W, TTG:W, CTT:M, CTC:D, CTA:R, CTG:G, TCT:S, TCC:S, TCA:H, TCG:S, CCT:P, CCC:N, CCA:N, CCG:Y, ACT:P, ACC:N, ACA:N, ACG:W, GCT:A, GCC:N, GCA:N, GCG:F, TAT:Y, TAC:P, TAA:*, TAG:*, CAT:Q, CAC:Y, CAA:D, CAG:V, AAT:F, AAC:G, AAA:K, AAG:N, GAT:D, GAC:G, GAA:E, GAG:G, TGT:P, TGC:L, TGA:*, TGG:W, CGT:R, CGC:C, CGA:M, CGG:H, AGT:S, AGC:F, AGA:G, AGG:S, GGT:G, GGC:W, GGA:V, GGG:Q. Rule 4: Translate sequentially until a stop (TAA, TAG, TGA → "*"). Rule 5: Output one-letter amino acid string. Rule 6: Stop at and include first "*". Rule 7: Unseen codons should not be assumed.
DNA Table Inference	Rule 1: DNA–protein pairs align codons (3 bp) to amino acids. Rule 2: Segment DNA into triplets from 5' and align to protein until "*" or end. Rule 3: Codons aligned to "*" are stop codons. Rule 4: First-codon→'M' pairs are start codons. Rule 5: Aggregate across examples; record all observed mappings. Rule 6: Include only codons seen. Rule 7: Build forward_table from all mappings, excluding stops. Rule 8: start_codons = all first codons mapped to 'M'. Rule 9: stop_codons = all codons aligned to '*'. Rule 10: Amino acids are single-letter codes including "*."
DNA Transform	Rule 1: Split input into 7-nt segments from 5'; last segment may be shorter. Rule 2: Reverse each 7-nt segment. Rule 3: Concatenate reversed segments to form output.

Table 9: Hypotheses Generated by GPT-4.1 for the DNA tasks

on chemistry-related tasks. In the molecule captioning task, Self-Consistency occasionally produced lower scores than the Implicit Inductive Reasoning baseline. While this strategy generates multiple hypotheses and applies them to derive answers, the resulting outputs were often fragmented or overly technical. For example, instead of producing full descriptive captions, as required by the task, the model frequently produced structural abbreviations or linkage names such as beta-D-Galp (1→4) beta-D-GlcpNAc (which are often part of the rule representations extracted by the model), omitting information about overall structure or functional roles. This indicates a misalignment between rule-based derivations and the task's requirement for holistic descriptions. In the reaction prediction task, Hypothesis Refinement also failed to deliver consistent improvements. Our analysis suggests that this was due to refined rules were not always effectively applied to the examples, and the selection of the "best" hypothesis depended solely on an automatic evaluator of prediction accuracy, which does not necessarily capture scientific plausibility.

Overall, these results suggest that the limitations of advanced reasoning strategies stem less from insufficient domain knowledge in base models than from structural mismatches between the strategies and the nuanced demands of the tasks.

C Experiment Details

C.1 Implementation Details

We run our experiments using API-based closed-source models, specifically claude-3-5-haiku-20241022, gpt-4.1-2025-04-14, and gemini-2.5-flash-preview-04-17. We implement our inference strategies in the OpenCompass framework. This allows us to perform inference in parallel at high rates. The explicit inductive reasoning is implemented via one-pass decoding, generating the hypothesis and applying it to the test example in one API call. Self-consistency is implemented by sampling multiple times using the same process as explicit inductive reasoning. For hypothesis refinement, we sample the hypothesis using the same general prompt in all tasks, except for DNA Translation where we ask the model to provide the specific codon-to-amino acid so that the hypothesis can be properly refined. For tasks in which the

Task	Hypothesis
Molecule Design	Rule 1: Identify required functional groups (e.g., diamine, aldehyde, etc.). Rule 2: Map biological role to known scaffolds (e.g., antineoplastic → stilbene). Rule 3: Choose core heterocycle per “derives from” (e.g., triazine). Rule 4: Decorate core with substituents to satisfy function and activity. Rule 5: Respect stereochemistry (e.g., [C@H] per natural enantiomer). Rule 6: For natural products, replicate known SMILES closely. Rule 7: Attach alkyl/aryl groups at correct positions. Rule 8: Output valid SMILES with rings, heteroatoms, charges.
Molecule Caption	Rule 1: Identify core ergot alkaloid and name (e.g., ergotaman). Rule 2: Describe substituents and positions (e.g., 12’-hydroxy). Rule 3: Note stereochemistry if differentiating isomers. Rule 4: Mention salts/derivatives (e.g., methanesulfonic acid salt). Rule 5: State biological origin or role if recognizable. Rule 6: Use “derives from” for parent relationships. Rule 7: Note naming conventions or historical context if relevant. Rule 8: Separate distinct features into clear sentences.
Reaction Prediction	Rule 1: Target N-heterocycle fused to benzene undergoes nucleophilic attack. Rule 2: Organometallics ([Li]CCCC, [H–]) add to carbonyl or halide. Rule 3: Bases ([NH_4^+], [OH–]) deprotonate or hydrolyze esters → amides/acids. Rule 4: Leaving groups replaced by nucleophiles forming C–X or C–C. Rule 5: Ester + nucleophile -> amide/ether. Rule 6: Most nucleophilic reagent reacts with most electrophilic center. Rule 7: Ignore spectator ions in final product. Rule 8: Grignard addition -> alcohol at addition site. Rule 9: Reductions ([H–]) convert carbonyls → alcohols/amines. Rule 10: On heteroaryl halide, nucleophile replaces halide on ring. Rule 11: Ethers/amides attach to aromatic systems via substitution/acylation. Rule 12: With both esters and amines, amide formation is preferred.
Name Prediction	Rule 1: Count all C atoms (including branches/rings). Rule 2: Count H via implicit valence rules. Rule 3: Count N, O, S, Si, halogens from SMILES. Rule 4: Include implicit Hs in aromatic rings per standard. Rule 5: Integrate substituent atoms without double-counting. Rule 6: Adjust H count for double/triple bonds. Rule 7: Write formula as C, H, then others alphabetically. Rule 8: Expand grouped atoms (e.g., O[Si](C)(C)C). Rule 9: Sum counts; check branching consistency. Rule 10: Format as [Element][count]… (e.g., C6H6O).

Table 10: Hypotheses Generated by GPT-4.1 for the Chemistry tasks

hypothesis can be translated into Python code, we prompt an LLM to generate the code. Otherwise, we prompt the LLM to apply a hypothesis to all in-context example inputs and do this to all the generated hypothesis. We used AI assistants to polish some of the text in this paper.

C.2 Prompts

Molecule Captioning As discussed in Section3.3, molecule captioning is an open-ended generation task, for which existing evaluations rely primarily on surface-level matching. To address this limitation, we design a dedicated prompt with fine-grained scoring criteria and employ an LLM to serve as the evaluator.

One-pass Self-Consistency To reduce the number of API calls and improve the efficiency of self-consistency, we design the prompt so that the model performs both rule induction and application to the test input within a single invocation.

Universal Majority Voting with Self-Consistency Given that the outputs of the chemistry and biology tasks in SIRBench-V1 are typically long and semantically complicated, basic majority voting mechanism often fails to identify a representative response, thereby diminishing the effectiveness of self-consistency. To address this, we adopt the universal self-consistency strategy(Chen et al., 2023), selecting the most semantically consistent response to form the final answer.

Hypothesis Refinement We provide the main prompts used in the hypothesis refinement process, including Hypothesis Induction, Hypothesis Application, Hypothesis Refinement, and Final Hypothesis Application.

D Complete Results on Chemistry Tasks

We provide the full results on Chemistry Tasks that reports all the metrics in table 11, table 13, and table 12.

Task	**Metric**	**Implicit Inductive Reasoning**	**Explicit Inductive Reasoning**	**Self-Consistency**	**Hypothesis Refinement**
Molecule Design	exact_match	0.17	0.23	0.23	**0.27**
	bleu	0.41	0.36	0.19	**0.71**
	levenshtein (↓)	70.87	84.70	173.47	**26.30**
	validity	0.70	0.77	**0.80**	0.70
	maccs_sims	0.81	0.75	0.84	**0.89**
	rdk_sims	**0.81**	0.69	0.69	0.76
	morgan_sims	0.62	0.64	0.66	**0.73**
	fcd (↓)	12.82	13.87	**12.46**	13.22
Molecule Caption	bleu2	0.20	0.22	**0.39**	0.24
	bleu4	0.14	0.15	**0.29**	0.17
	rouge_1	0.33	0.24	**0.48**	0.40
	rouge_2	0.18	0.12	**0.29**	0.23
	rouge_l	0.25	0.19	**0.38**	0.31
	meteor_score	0.39	0.23	**0.44**	0.42
	LLM as judge	67.70	54.00	69.70	**72.70**
Reaction Prediction	accuracy	**44.44**	19.23	20.83	28.00
smiles2formula	accuracy	0.00	0.00	0.00	0.00
smiles2iupac	accuracy	0.00	0.00	0.00	0.00
iupac2smiles	accuracy	**14.29**	4.55	0.00	4.17
iupac2formula	accuracy	0.00	**6.67**	3.33	3.33

Table 11: Performance of the **Claude-3.5-Haiku** on Chemistry Tasks

Task	**Metric**	**Implicit Inductive Reasoning**	**Explicit Inductive Reasoning**	**Self-Consistency**	**Hypothesis Refinement**
Molecule Design	exact_match	**0.30**	0.20	0.20	0.23
	bleu	**0.75**	0.71	0.70	**0.75**
	levenshtein (↓)	25.37	27.93	26.37	**24.03**
	validity	0.87	**1.00**	0.93	0.93
	maccs_sims	**0.92**	0.87	0.91	0.87
	rdk_sims	0.80	0.74	**0.82**	0.78
	morgan_sims	**0.75**	0.69	0.72	0.67
	fcd (↓)	8.16	**7.08**	7.97	7.43
Molecule Caption	bleu2	0.42	**0.49**	**0.49**	0.20
	bleu4	0.32	0.38	**0.39**	0.15
	rouge_1	0.55	0.55	**0.57**	0.38
	rouge_2	0.36	0.38	**0.39**	0.24
	rouge_l	0.44	0.46	**0.48**	0.31
	meteor_score	**0.57**	0.52	0.54	0.48
	LLM as judge	**66.30**	59.00	65.70	**66.30**
Reaction Prediction	accuracy	22.22	17.86	25.00	**32.14**
smiles2formula	accuracy	**13.33**	6.67	10.00	10.00
smiles2iupac	accuracy	0.00	0.00	0.00	0.00
iupac2smiles	accuracy	**17.39**	4.35	5.00	13.04
iupac2formula	accuracy	**23.33**	13.33	**23.33**	**23.33**

Table 12: Performance of the **GPT-4.1** on Chemistry Tasks

Task	**Metric**	**Implicit Inductive Reasoning**	**Explicit Inductive Reasoning**	**Self-Consistency**	**Hypothesis Refinement**
Molecule Design	exact_match	**0.33**	0.27	0.27	0.20
	bleu	0.73	**0.79**	**0.79**	0.76
	levenshtein (↓)	27.90	25.27	**22.50**	26.67
	validity	0.80	0.77	**0.90**	0.73
	maccs_sims	**0.95**	0.94	0.94	0.81
	rdk_sims	**0.89**	0.86	0.87	0.82
	morgan_sims	**0.85**	0.77	0.80	0.72
	fcd (↓)	8.19	8.89	**6.26**	10.56
Molecule Caption	bleu2	0.49	**0.54**	0.51	0.42
	bleu4	0.38	**0.43**	0.41	0.33
	rouge_1	0.57	**0.61**	**0.61**	0.52
	rouge_2	0.38	**0.42**	0.41	0.35
	rouge_l	0.47	**0.50**	0.49	0.43
	meteor_score	0.55	**0.59**	**0.59**	0.52
	LLM as judge	63.30	67.70	**70.00**	65.70
Reaction Prediction	accuracy	**54.17**	34.78	39.29	32.14
smiles2formula	accuracy	**30.00**	20.00	**30.00**	16.67
smiles2iupac	accuracy	0.00	0.00	**3.33**	0.00
iupac2smiles	accuracy	20.00	40.00	**53.85**	52.94
iupac2formula	accuracy	70.00	60.00	**73.33**	66.67

Table 13: Performance of the **Gemini-2.5-Flash** on Chemistry Tasks

LLM-as-Judge Evaluation of Molecule Captioning:
You are an expert molecular biologist.
Below is a SMILES string representing a molecule: `{smiles}`
Here is a reference description of the molecule: `{gt}`
Here is a predicted description of the same molecule: `{pred}`
Your task is to evaluate the **predicted** description **only** based on its scientific quality compared to the reference.
You must assign a **score from 1 to 10** based on the following criteria:

- **Score 10**: Nearly perfect — scientifically precise, complete, and fluent. Matches all key aspects of the reference (e.g., functional groups, chemical class, derivation, roles).
- **Score 8–9**: Very good — minor omissions or slight rewording, but the core structure-level and functional meaning is intact.
- **Score 6–7**: Reasonable — generally correct but may lack specific details (e.g., derivation or one functional role). Possibly vague phrasing.
- **Score 4–5**: Partial — captures the general category or one function but omits multiple important details or shows misunderstanding in phrasing.
- **Score 2–3**: Poor — vague, generic, or scientifically weak. May refer to the wrong compound type or confuse structural features.
- **Score 1**: Completely incorrect or irrelevant.

Only output a **single line** in the following format: `Score: [1-10]`

One-pass Self-Consistency:
Below is a full prompt about the reasoning task, which includes the ICL examples and a new test case. **Your task is:**

1. Read the full prompt to understand the task and identify: 1) the example input-output pairs 2) the specific input question to answer.
2. Analyze these example pairs and generate a series of rules that explains how each input is transformed to its corresponding output.
3. Then, apply those rules to the final test question and output the answer.
4. Return your answer in the following format:

```
<rules>
Rule 1: ...
Rule 2: ...
Rule 3: ...
...
</rules>

<answer>
{{your answer}}
</answer>
```

Full prompt: `{full_prompt}`

Universal Majority Voting with Self-Consistency:
You are given a reasoning task prompt and multiple candidate responses to the question in that prompt. **Your task is:**

1. Read the full prompt carefully to understand the question being asked.
2. Examine all the candidate responses and determine whether any of them form a majority consensus.
    - A majority exists if **any single response appears more than any other** (either verbatim or semantically equivalent).
    - In case of a tie (e.g., all responses differ or two responses appear with equal frequency), consider that no majority exists.
3. If a majority exists, return that response as the final answer.
4. If no majority exists, then select the **most reasonable and task-appropriate** response based on the prompt.

Candidate responses: `{responses}`
Full prompt: `{full_prompt}`
Return your final answer using exactly the following format:

```
majority_found: [yes or no]
selected_response: {full response content}
```

Example:

```
majority_found: yes
selected_response: This is the most common (or semantically equivalent)
response and correctly answers the question.
```

Hypothesis Induction Prompt
Below is a full prompt about the reasoning task, which includes the ICL examples that you should learn from. **Your task is:**

1. Read the full prompt to understand the task and identify the example input-output pairs.
2. Analyze these example pairs and generate a series of rules that explains how each input is transformed to its corresponding output.
3. Provide as much detail as possible in the rules, such as elaborating on the specific mapping.{note}
4. Return your rules in the following format (each rule on its own line):

```
<hypothesis>
Rule 1: ...
Rule 2: ...
Rule 3: ...
...
</hypothesis>

Full prompt:
{full_prompt}
```

Hypothesis Application Prompt (General)
Task Description: task_description
Please apply the given hypothesis to the given list of inputs. Ensure that you provide the actual output for each input. Do not give a program, partial output, or placeholder.
Hypothesis: hypothesis
Input: icl_in
Format your output as follows:

```
<output>
Output 1: ...
Output 2: ...
...
</output>
```

DNA Table Prompt
Below is a full prompt about the reasoning task, which includes the question that you should give the corresponding answer. **Your task is:**

1. Read the full prompt to understand the task and identify the specific input question to answer.
2. Based on your understanding of the given rules, generate the corresponding output for the question.

Rules: hypothesis
Full prompt: x
Enclose your answer with `<answer></answer>` tags.

DNA Translation/Transformation as Python Code Prompt

Convert the following hypothesis into a Python function called `apply` that takes a string input and returns the transformed output. The function should implement the rules described in the hypothesis. Make sure to handle all the transformations correctly.

Task Description: self.task_description

Hypothesis: hypothesis

Your function should follow this template:

```
def apply(input_str):
    # Implementation based on the hypothesis rules
    # ...
    return result
```

Return ONLY the Python code without any explanation or markdown formatting.

Hypothesis Refinement Prompt
You are given a candidate hypothesis that attempts to explain how each input is transformed into its output. A hypothesis consists of rules that explain how the inputs are mapped to the outputs. Your goal is to revise this hypothesis so it fully accounts for any discrepancies. You may add new rules, modify existing ones, or remove inaccurate ones. You can also propose a completely new hypothesis.
Context: self.task_description
Current Hypothesis: hypothesis
Input: icl_in
Model Output: generated_output
Expected Output: expected_output
Steps:

1. List the exact differences between Model Output and Expected Output.
2. For each difference, identify which existing rule (if any) fails to cover it.
3. Revise existing rules or introduce new rules to fix these gaps.
4. Ensure the rules clearly state how the input is mapped into output in a detailed manner.{note}

Output only the refined hypothesis—do not solve the original task.
Format your output as follows:

```
<new_hypothesis>
Rule 1: ...
Rule 2: ...
Rule 3: ...
...
</new_hypothesis>
```

Final Hypothesis Application Prompt
Below is a full prompt about the reasoning task, which includes the question that you should give the corresponding answer. **Your task is:**

1. Read the full prompt to understand the task and identify the specific input question to answer.
2. Based on your understanding of the given rules, generate the corresponding output for the question.

Rules: hypothesis
Full prompt: x
Enclose your answer with `<answer></answer>` tags.