LogiDynamics: Unraveling the Dynamics of Inductive, Abductive and Deductive Logical Inferences in LLM Reasoning

Tianshi Zheng[1], **Jiayang Cheng**[1], **Chunyang Li**[1], **Haochen Shi**[1], **Zihao Wang**[1], **Jiaxin Bai**[1]
Yangqiu Song[1], **Ginny Y. Wong**[2], **Simon See**[2]
[1]Department of Computer Science and Engineering, HKUST, Hong Kong SAR, China
[2]NVIDIA AI Technology Center (NVAITC), NVIDIA, Santa Clara, USA
tzhengad@connect.ust.hk

Abstract

Modern large language models (LLMs) employ diverse logical inference mechanisms for reasoning, making the strategic optimization of these approaches critical for advancing their capabilities. This paper systematically investigate the **comparative dynamics** of inductive (System 1) versus abductive/deductive (System 2) inference in LLMs. We utilize a controlled analogical reasoning environment[1], varying modality (textual, visual, symbolic), difficulty, and task format (MCQ / free-text). Our analysis reveals System 2 pipelines generally excel, particularly in visual/symbolic modalities and harder tasks, while System 1 is competitive for textual and easier problems. Crucially, task format significantly influences their relative advantage, with System 1 sometimes outperforming System 2 in free-text rule-execution. These core findings generalize to broader in-context learning. Furthermore, we demonstrate that advanced System 2 strategies like hypothesis selection and iterative refinement can substantially scale LLM reasoning. This study offers foundational insights and actionable guidelines for strategically deploying logical inference to enhance LLM reasoning.

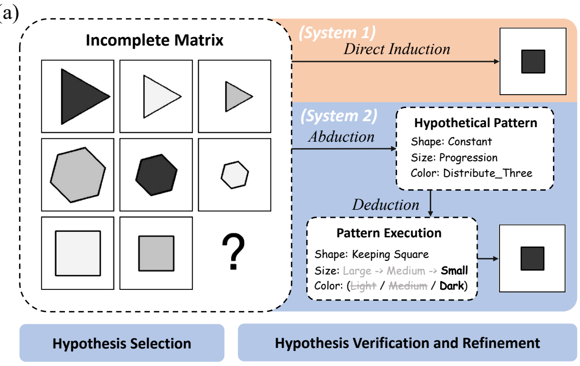
"It is not enough to have a good mind; the main thing is to use it well."

— René Descartes

1 Introduction

Logical Inference[2] is the reasoning process of deriving conclusions from known premises (Copi and Cohen, 1990; Johnson-Laird, 2010). It primarily categorizes into *deductive inference* — where conclusions follow with logical necessity from premises, and *inductive inference* — where conclusions serves as general rules derived from specific instances (Salmon, 1984). While the introduction of *abductive inference* (Peirce, 1958; Frankfurt, 1958) serves as a third perspective, denoting the process of forming an explanatory hypothesis from an observation requiring explanation. Logical inference plays a crucial role in artificial intelligence, scientific research, and philosophy, where rational decision-making and hypothesis formation are foundational (Hempel and Oppenheim, 1948; Harman, 1965; Reiter, 1987).

(a)



(b)

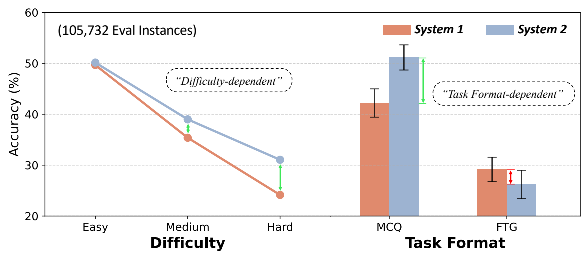


Figure 1: (a) An illustration of System 1 and System 2 logical inference pipelines in RAVEN's progressive matrix. (b) General **comparative dynamics** between System 1 and System 2 pipelines in all experiments.

Different logical inference pipelines can be applied in solving the same reasoning task. Figure 1(a) illustrates an example of Raven's Progressive Matrices (Raven, 1938; Zhang et al., 2019), where

[1]https://github.com/HKUST-KnowComp/LogiDynamics
[2]The term 'inference' encompasses multiple interpretations across different disciplines. This paper employs the term strictly within its logical trichotomy: deductive, inductive, and abductive inference, as defined in (Flach and Kakas, 2000).

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 20722–20743
November 4-9, 2025 ©2025 Association for Computational Linguistics

the missing element in the 3×3 matrix is inferred through the common patterns among different rows. There are two approaches to solving this problem: 1) directly inferring the missing element from the observed elements in the matrix, and 2) explicitly identifying the common patterns across rows, then deductively applying these patterns to determine the missing element in the last row. The former is driven by inductive inference and features fast, intuitive, pattern-recognition guided reasoning. The latter consists of abductive and deductive inference, featuring slower but more deliberate analysis. These approaches correspond to System 1 and System 2 thinking, respectively (Kahneman, 2011).

Research on large language models (LLMs) has explored the logical inference pipelines employed by LLMs for solving a wide range of tasks. Qiu et al. (2024) and Wang et al. (2024) have demonstrated the effectiveness of the System 2 approach in various inductive reasoning datasets such as ARC (Chollet, 2019) and its variants (Kim et al., 2022; Xu et al., 2023). He et al. (2024) highlighted the potential of System 2 logical inference in the reasoning workflow of LLM-based agents. While Liu et al. (2024) compared both System 1 and System 2 approaches in several in-context learning tasks, pointing out the inconsistency of their relative performances across datasets. Nevertheless, all prior studies leave an open question: *When* and *how* can System 1 and System 2 logical inference pipelines be effectively leveraged to enhance LLM reasoning?

To address this intricate question, we systematically investigate the **comparative dynamics** of System 1 and System 2 pipelines within LLM reasoning tasks, specifically examining the contingency of their performance preferences on task attributes such as modality, difficulty, and task format. First, we build a fully controllable evaluation environment using analogical reasoning tasks. The environment is controlled in three dimensions: 1) *Modality*: The data covers textual (word/phrase), visual (images), and symbolic modalities. 2) *Difficulty*: All tasks are labeled with relative difficulty levels (easy, medium, and hard). 3) *Task Format*: For each question, we provide two task formats: multiple-choice questions (MCQ) or free-text generation (FTG) format.

With experiments in 10 modern LLMs (and MLLMs), **we discover several key findings**:

- **Modality-dependent**: System 2 logical inference shows superior performance in *visual* and *symbolic* tasks, while System 1 performs comparably in *textual* tasks.
- **Difficulty-dependent**: System 2 logical inference is more advantageous in *harder* tasks, while System 1 achieve comparable performance in *easier* tasks.
- **Task Format-dependent**: For tasks involving explicit rule execution, System 1 logical inference outperforms System 2 in *FTG* format, but underperforms in *MCQ* format.

To verify the generalizability of our findings, we conduct further experiments in the List Function dataset (Rule, 2020) and SALT dataset (ours), where we observe similar comparative dynamics in difficulty and task format. We argue that **our findings can be generalized to broader in-context learning (ICL) tasks** where: 1) the few-shot demonstrations are presented in Input-Output format, and 2) the mapping function between input and output can be explicitly defined.

Furthermore, we explored the effects of more sophisticated System 2 logical inference pipelines, including hypothesis selection, hypothesis verification, and refinement. Using these paradigms, LLMs demonstrate significant performance improvements as the number of inference tokens increases. We show that, with sufficient computational resources, **LLMs under logical inference scaling achieve performance comparable to state-of-the-art Long-CoT reasoning models**. This highlights the potential of scaling inference through advanced System 2 logical inference pipelines.

This work makes several key contributions to understanding and improving LLM reasoning capabilities from a logical inference perspective:

1. We provide a **systematic evaluation environment** to compare logical inference paradigms across controlled dimensions. (§3)
2. We present **rich findings as clear guidelines** for leveraging different inference approaches based on task characteristics. (§4)
3. We validate our findings' **generalizability to broader in-context learning tasks**. (§5)
4. We highlight the potential to **scale up LLM reasoning** using advanced System 2 logical inference paradigms. (§6)

Collectively, these contributions establish a foundation for future research on enhancing LLM reasoning through optimized logical inference strategies.

2 Preliminaries

2.1 Analogical Reasoning

Analogical reasoning is a fundamental aspect of cognitive intelligence (Gentner et al., 2001). It involves inferring a missing element in a target domain according to relational structures from a source domain. Formally, given a source pair (A, A') and an incomplete target pair (B, x), where A and A' have an implicit relational pattern P, the goal is to infer x that have the same relational pattern P with B. This task can be defined as:

$$B' = \arg\max_{x \in \mathcal{X}} \text{sim}_P((A, A'), (B, x)),$$

where sim_P measures the consistency of the relational pattern P between the source pair (A, A') and the candidate target pair (B, x), and $\mathcal{X}$ represents the set of all possible candidates for B'. The complete analogy is denoted as $A : A' :: B : B'$. For instance, given the source pair $(\textit{sun}, \textit{planet})$ and the incomplete target pair $(\textit{nucleus}, x)$, we can infer $x = \textit{electron}$ by identifying the pattern P as *orbital relationship*.

The task of analogical reasoning is particularly well-suited for our investigation for several reasons: 1) it offers a well-defined task structure while encompassing diverse data modalities, 2) it is compatible with a variety of logical inference pipelines, and 3) it is considered out-of-distribution for the training data of LLMs, enabling a robust evaluation of their reasoning capabilities under generalization (Stevenson et al., 2024).

2.2 Logical Inference Pipelines

In the main experiment, we compare three logical inference pipelines: direct induction, abduction + deduction, and automatic inference. More sophisticated pipelines involving hypothesis selection, verification, and refinement are discussed in the scaling experiments in Section 6. Detailed prompt templates are provided in Appendix D.

Direct Answering as Inductive Inference Inductive inference is often associated with fast, intuitive reasoning in cognition (Cohen, 1982). Similar to Liu et al. (2024), we regard the direct answering of LLMs as a form of inductive inference, representing their System 1 logical inference pipeline.

Dataset			Difficulty			Total
Task	Modality	Benchmark	Easy	Medium	Hard	
Analogy	Textual	E-KAR	317	435	496	1248
	Visual	VASR	455	572	320	1347
	Symbolic	RAVEN	402	462	395	1259
General ICL	Math/Code	List Function	432	423	395	1250
	Textual	SALT	400	400	400	1200
Total			2006	2292	2006	**6304**

Table 1: Dataset statistics across modalities and difficulty levels. Details of general in-context learning tasks (List Function and SALT) are introduced in Section 5.

Abductive and Deductive Inference With this System 2 pipeline, task completion is decomposed into two steps. First, LLMs are required to abductively infer the hypothetical pattern P_h based on the source pair(s). Then, they deductively apply this pattern to the incomplete target pair as $B \xrightarrow{P_h} B'$.

Zero-shot CoT as Automatic Inference The reasoning process observed in zero-shot CoT (Chain-of-Thought) (Wei et al., 2023), which we term "Automatic Inference" for the purpose of this paper, demonstrates an inherent logical inference capability acquired during instruction-tuning or alignment stages. Therefore, we included the "Automatic Inference" in our comparison for reference.

3 Evaluation Environment

In this section, we introduce our evaluation environment of analogical reasoning, providing details on the settings for each control dimensions.

3.1 Modality

Exploring diverse data modalities is crucial for obtaining comprehensive insights. To this end, we selected three analogical reasoning tasks across different modalities. **E-KAR** (Chen et al., 2022) consists of human-curated analogy questions between word pairs (or sets), where analogies are determined by shared ontological relationships between words. **VASR** (Bitton et al., 2022) comprises human-annotated analogical questions between image pairs, where analogies are determined by shared semantic transitions between images. **RAVEN** (Raven, 1938; Zhang et al., 2019; Hu et al., 2022) generates symbolic matrices using attributed stochastic image grammar (A-SIG), where analogies are determined by shared attribute shifts among rows. To enhance comprehension in large language models, we adopt the abstracted version proposed by Hu et al. (2023), which tokenizes the

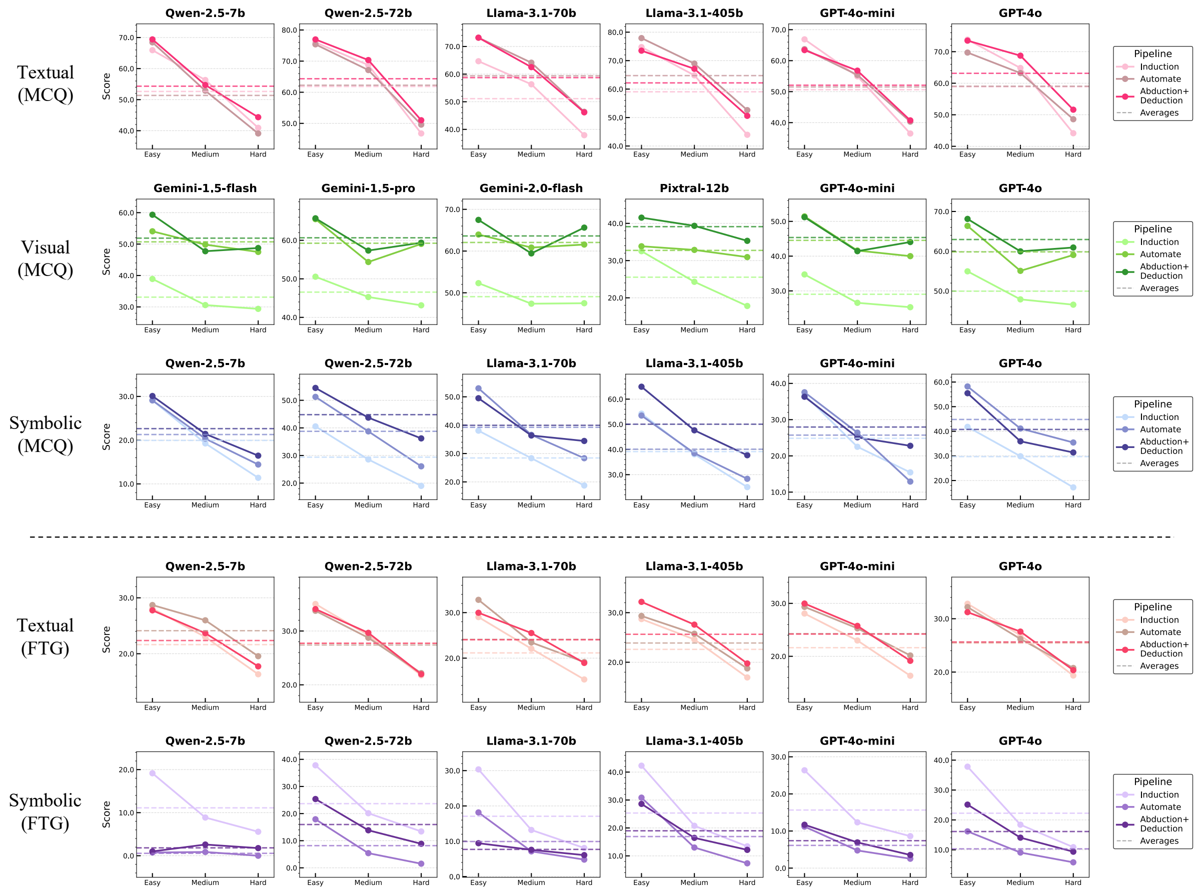


Figure 2: LLM performances (in Accuracy %) in our evaluation environment under different reasoning pipelines.

matrix images into symbolic vectors.

3.2 Difficulty

Task difficulty, while a key determinant of thinking styles (Phillips et al., 2016), is largely overlooked in research on reasoning paradigms in LLMs. To address this, we conducted difficulty annotations for all three datasets. In analogical reasoning involving real-world data, difficulty is often measured by the semantic distance between analogy pairs (Vendetti et al., 2012; Jones et al., 2022). For E-KAR, we compute the semantic distance between word pairs using **FastText** embeddings (Bojanowski et al., 2017), which are more suitable than Word2Vec (Mikolov et al., 2013a) or BERT (Devlin et al., 2019), as the word pairs exhibit morphological variations but lack contextual dependencies. For VASR, we calculate the distance between **VGG** encodings (Simonyan and Zisserman, 2015) to account for both semantic and graphical features. For RAVEN, task complexity is defined by the number of **attribute variations** across the columns. The statistics of our datasets across different modalities and difficulty levels are presented in Table 1. Further details about our difficulty annotation process are provided in Appendix B.

3.3 Task Format

The task format also serves as an important factor influencing reasoning performance (Ribeiro et al., 2018; Zong et al., 2024). We conducted experiments separately under two task formats[3]: multiple-choice questions (MCQ) and free-text generation (FTG), aiming to achieve a more comprehensive perspective in our exploration.

4 Main Experiment Results and Analysis

We evaluated 10 modern LLMs / MLLMs (details provided in Appendix A) within our exploration environment. The experimental results are presented in Figure 2. Across the entire environment, the tested LLMs achieved an overall average performance of only **35.4%**, demonstrating that our datasets effectively stress-test the real reasoning abilities of LLMs rather than simply retrieving from memorization. Furthermore, the substantial

[3]For the visual dataset, we evaluated only in the MCQ format for feasibility.

(a) Modality (Task Format = MCQ)

Pipeline	Modality		
	Textual	Visual	Symbolic
Induction	55.70	38.88	28.58
Automatic	58.05	51.52	34.99
Abduction+Deduction	**59.13**	**53.93**	**37.69**
System 2 Advantage	+6.16%	+38.73%	+31.86%

(b) Difficulty (Task Format = MCQ)

Pipeline	Difficulty		
	Easy	Medium	Hard
Induction	51.48	41.93	31.48
Automatic	58.12	48.23	40.02
Abduction+Deduction	**59.68**	**49.76**	**43.20**
System 2 Advantage	+15.92%	+18.68%	+37.20%

(c) Task Format

Pipeline	Textual		Symbolic	
	MCQ	FTG	MCQ	FTG
Induction	55.70	23.36	28.58	**19.18**
Automatic	58.05	24.89	34.99	8.67
Abduction+Deduction	**59.13**	**24.93**	**37.69**	11.33
System 2 Advantage	+6.16%	+6.74%	+31.86%	-40.93%

Table 2: Comparative dynamics of different logical inference pipelines in our evaluation environment, controlled by *modality*, *difficulty*, and *task format*. Performances (in Accuracy %) are averaged across all LLMs. "System 2 Advantage" denotes the relative improvements of abduction + deduction pipeline over direct induction.

performance gaps across difficulty levels validate the effectiveness of our difficulty annotations. Generally, the abduction + deduction pipeline outperforms direct induction, while automatic inference falls between the two pipelines in most scenarios.

To better illustrate the comparative dynamics between different logical inference pipelines, we present the consolidated results controlled by each dimension in Table 2. From these results, we observe the key findings as follows:

Findings 1: The comparative advantages of the System 2 logical inference pipeline are modality-dependent. As shown in Table 2(a), the abduction + deduction pipeline substantially outperforms direct induction in visual and symbolic tasks, with relative improvements of 38.73% and 31.86%, respectively. However, in textual tasks, direct induction achieves comparable performance, trailing behind by only 6.16%.

Findings 2: The comparative advantages of the System 2 logical inference pipeline are difficulty-dependent. Based on Table 2(b), the abduction + deduction pipeline outperforms direct induction by 37.20% on hard questions, while the performance gap reduces to 18.68% and 15.92% on medium and easy questions, respectively.

Findings 3: The System 2 logical inference pipeline falls short in free-text generation format when the task requires explicit rule execution. Results from Table 2(c) reveal a *noteworthy inconsistency*: in textual tasks, the advantage of the System 2 pipeline remains the same across task formats. However, in symbolic tasks (i.e., RAVEN), the System 2 pipeline severely underperforms direct induction in the free-text generation format, which **sharply contrasts** with its advantage in the multiple-choice question format.

Interpretation of Findings 3: To investigate the underlying mechanism leading to the limitation of System 2 logical inference in free-text generation, we conducted further analyses to decouple the performance of abduction and deduction (detailed in Appendix E). We identified the following explanations for this task format sensitivity:

- The precise generation of complex rules is challenging for most LLMs, as evidenced by the poor pattern inference accuracy compared to pattern execution (Table 8).
- Implicit pattern matching may be more effective in this case, as employed by direct induction. However, in the System 2 pipeline, lengthy rationales disrupt the well-structured few-shot patterns essential for in-context learning, thereby rendering implicit learning ineffective (Table 9).
- For multiple-choice questions, the System 2 pipeline can better infer patterns, as the answer space is reduced to a few candidates. It may also occasionally leverage reasoning shortcuts to improve performance (Geirhos et al., 2020; Zong et al., 2024) — an advantage that cannot be employed in direct induction.

As a result, the abduction + deduction pipeline tends to favor the MCQ format when addressing problems that require explicit rule execution, whereas, under the FTG format, direct induction demonstrates a surprising advantage.

5 Generalization Experiment

To further assess the generalizability of our findings, we extend the scope from analogical reason-

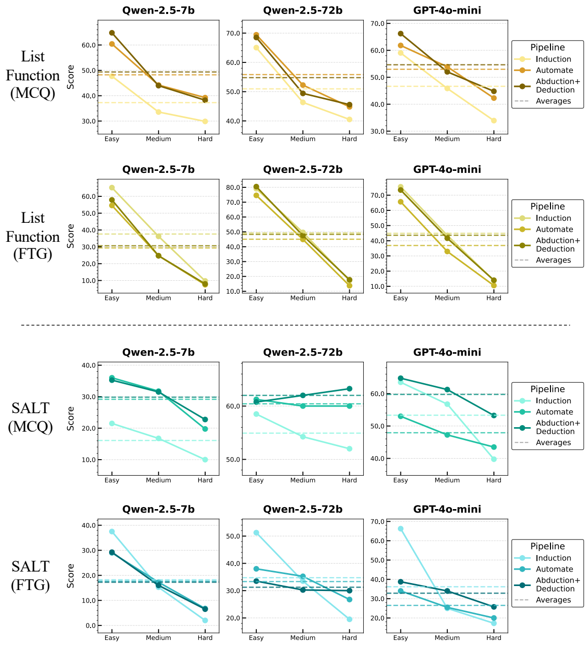


Figure 3: LLM performances (in Accuracy %) in List Function and SALT under different reasoning pipelines.

ing to general in-context learning tasks. Specifically, we formally define our target task scope using the following constraints: 1) The task requires generating output y from input x, based on n-shot demonstrations $D = [(x_1, y_1), \ldots, (x_n, y_n)]$. 2) The input-output function $y = f(x)$ can be explicitly defined. We conduct generalization experiments on two in-context learning datasets, both of which require explicit rule execution.

List Function (Rule, 2020) takes lists of integers as input and maps them to output lists using 250 predefined transition functions. In this task, LLMs must infer the underlying function from provided demonstrations (input-output pairs) and apply it to new input lists. The difficulty of the task is determined by the complexity of the transition functions.

SALT (**S**yntax-aware **A**rtificial **L**anguage **T**ranslation) is a machine translation benchmark that we developed to address key limitations in existing datasets. Unlike benchmarks such as SCAN (Higgins et al., 2018) and Kalamang (Tanzer et al., 2024), SALT introduces diverse syntactic shifts (e.g., inversion of semantic unit order) while rigorously mitigating data leakage—a common issue in low-resource machine translation benchmarks. The task difficulty is determined by the complexity of the syntactic structures, enabling fine-grained evaluation of model performance across varying levels of linguistic challenge. Details of the SALT

(a) List Function

Pipeline	Difficulty			Task Format	
	Easy	Medium	Hard	MCQ	FTG
Induction	65.26	42.53	24.18	44.96	**43.92**
Automatic	64.42	42.16	26.35	52.35	37.09
Abduction+Deduction	**68.55**	**43.21**	**28.06**	**52.93**	40.85
System 2 Advantage	+5.04%	+1.60%	+16.06%	+17.73%	-6.98%

(b) SALT

Pipeline	Difficulty			Task Format	
	Easy	Medium	Hard	MCQ	FTG
Induction	**49.75**	33.58	23.42	41.44	**29.72**
Automatic	41.88	36.17	29.46	45.83	25.83
Abduction+Deduction	43.71	**39.17**	**33.58**	**50.53**	27.11
System 2 Advantage	-12.14%	+16.63%	+43.42%	+21.92%	-8.79%
Average	-3.55%	+9.11%	+29.74%	+19.83%	-7.88%

Table 3: Comparative dynamics of different logical inference pipelines in List Function and SALT. Performances (in Accuracy %) are averaged across all LLMs.

dataset are provided in Appendix C.

The results of the generalization experiments are illustrated in Figure 3, with the consolidated findings presented in Table 3. Across both datasets, we observed patterns similar to those in our evaluation environment in analogy: The advantage of the System 2 logical inference pipeline increases substantially as task difficulty rises. While the two pipelines exhibit contrasting task preferences between the MCQ and FTG format. Consequently, we demonstrate that **our findings are generalizable to broader in-context learning tasks** where the input-output function is explicitly defined.

6 Scaling-up System 2 Logical Inference

Beyond the basic processes of abductive hypothesis generation and deductive execution (which form the core of our System 2 pipeline), more sophisticated logical inference strategies can be employed to tackle complex tasks and further enhance System 2 reasoning. We introduce two inference methodologies in philosophy and connect them to the logical inference pipelines of LLMs.

6.1 Liptonian and Holmesian Inference

Liptonian Inference (Lipton, 2000) provides a widely recognized modern account of IBE (Inference to the Best Explanation). It characterizes the process of selecting the most explanatory hypothesis from a set of candidates based on its capacity to best account for the observed evidence. In LLM reasoning, this corresponds to the parallel sampling of multiple hypotheses, followed by hypothesis se-

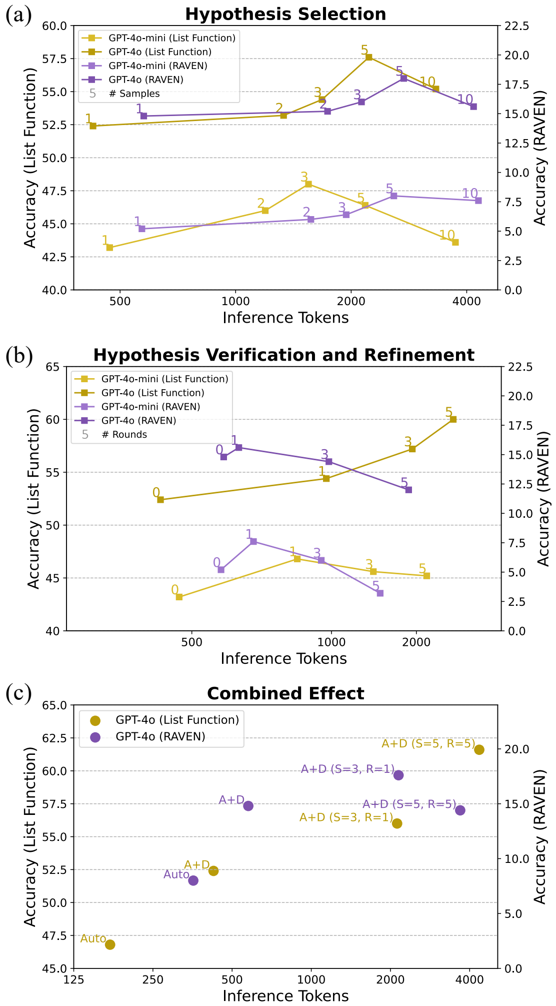


Figure 4: Effect of hypothesis selection, verification and refinement on LLM performances (in Accuracy %).

lection as a precursor to the final deductive execution. In our experiment, we evaluated the effectiveness of **hypothesis selection** across sampling sizes ranging from 1 to 10.

Holmesian Inference (Bird, 2005) provides an alternative model to Liptonian, emphasizing hypothesis verification rather than selection. Inspired by Sherlock Holmes's famous dictum, it involves systematically eliminating all but one hypothesis to ensure that the remaining one is necessarily true. In LLM reasoning, this can be simulated through iterative verification and refinement (regeneration) of hypotheses, where candidate outputs are repeatedly evaluated and improved. In our experiment, we investigated **hypothesis verification and refinement** across iteration rounds up to 5.

6.2 Scaling Performances

The experimental results of hypothesis selection, verification and refinement are presented in Figure 4 (a) and 4 (b). In hypothesis selection, we observe clear improvements in sampling sizes from 1 to 5. However, the performance starts to decrease when the sampling size increases to 10, as the diversity of the sampled hypotheses begins to saturate, and the selection process also becomes less effective with a longer context. In terms of hypothesis verification and refinement, the saturation of improvements was reached after one round of verification, except for GPT-4o in the List Function, where positive improvements were observed in every additional round of verification. This interesting inconsistency can be explained as follows: **1) Stronger LLMs lead to better verification quality.** Compared to the consistent improvements observed in GPT-4o, GPT-4o-mini did not exhibit similar enhancements, as its ability to detect incorrect hypotheses is also weaker. **2) Well-formed hypothesis formats make refinements easier.** The improvement seen in the List-Function dataset (where hypotheses are written in Python code) does not hold for the RAVEN dataset (where hypotheses are presented in free text). A better hypothesis format may also enhance the effectiveness of proofreading or maintaining the validity of existing hypotheses.

Figure 4 (c) illustrates the combined effect of the two scaling strategies. In both datasets, GPT-4o demonstrates considerable performance improvements as the number of inference tokens increases. For instance, performance of GPT-4o in the List-Function dataset improved from **46.8%** to **61.6%**, consuming **25×** more inference tokens compared to automatic inference. **This underscores the potential of scaling up LLM reasoning through System 2 logical inference pipelines.**

6.3 Discussions on Large Reasoning Models

Recent advances in large reasoning models (LRMs), such as o1 (OpenAI, 2024) and Deepseek-R1 (DeepSeek-AI et al., 2025), have demonstrated impressive performance in mathematical and code reasoning tasks. LRMs emerge strong self-reflec- tion abilities during their reinforcement learning stage, driven by rule-based rewards. From our exploration, LRMs exhibit two noteworthy characteristics within our task domain (in-context learning with explicit input/output functions): 1) **LRMs emulate an "iterative holmesian inference"** by engaging in repeated cycles of hypothesis generation and verification. 2) **The number of inference tokens (rounds of iterative hypothesis generation) increases substantially as task difficulty rises**.

Model	Inference Tokens (# Rounds)			Accuracy
	Easy	Medium	Hard	
Deepseek-R1	2174.5 (3.9)	3353.1 (5.0)	5935.9 (6.5)	69.2
o1-mini	1345.5 (2.6)	2229.8 (3.2)	4188.0 (3.5)	69.6
o1	1949.1 (2.7)	3233.0 (3.3)	6995.7 (5.5)	**77.2**
o3-mini	1184.3 (2.5)	2126.3 (3.0)	5328.7 (6.2)	<u>76.8</u>
Deepseek-V3	989.0	1261.1	1260.9	57.2
+Sys2 Scaling (low)	1758.0 (2.4)	2124.4 (2.5)	2618.9 (2.7)	65.2
+Sys2 Scaling (high)	2356.8 (2.7)	2985.3 (2.9)	4308.0 (3.8)	69.6

Table 4: Performance of LRMs and LLMs with adaptive logical inference scaling on the List Function dataset.

Nevertheless, can short-CoT LLMs achieve comparable performance by scaling up System 2 logical inference? To answer this question, we conducted experiments on Deepseek-V3 (DeepSeek-AI et al., 2024), employing adaptive logical inference scaling under *low* and *high* computational consumptions (details in Appendix F), where the model autonomously determined the number of iteration within a set limit. As illustrated in Table 4, under high consumptions, Deepseek-V3 **demonstrates a similar inference scaling effect in difficulty and achieves comparable performance to LRMs**.

7 Related Work

7.1 Logical Inference in Language Models

Abductive Inference In the era of pre-trained language models, α**-NLI** (Bhagavatula et al., 2020) introduced abductive reasoning to commonsense reasoning, where plausible explanations are inferred from observations. Subsequent works proposed various techniques to enhance this capability (Qin et al., 2021; Kadiķis et al., 2022; Chan et al., 2023), including extensions to uncommon scenarios focusing on rare but logical explanations (Zhao et al., 2024). Unlike real-world data in commonsense reasoning, benchmarks like **ProofWriter** (Tafjord et al., 2021) evaluate formal abductive reasoning within semi-structured texts with explicit logical relationships. Recent studies have explored LLMs in more challenging open-world reasoning contexts (Zhong et al., 2023; Del and Fishel, 2023; Thagard, 2024). Beyond natural language inference, abductive reasoning has also been examined in graph-based modalities for commonsense and event knowledge (Du et al., 2021; Bai et al., 2024).

Deductive and Inductive Inference Deductive inference is extensively studied in transformers (Clark et al., 2020; Han et al., 2024; Zheng et al., 2025b) with natural language rule-based reasoning tasks. Saparov et al. (2023) evaluate LLMs' deductive reasoning in out-of-distribution settings, emphasizing challenges with longer proofs and complex logic. Inductive inference is explored through datasets like **EntailmentBank** (Dalvi et al., 2022), where models construct step-by-step entailment trees to explain answers. Meanwhile, recent studies have demonstrated emergent inductive reasoning abilities in LLMs (Zheng et al., 2025a; Li et al., 2025; Fan et al., 2025).

7.2 Analogical Reasoning

The study of analogical reasoning in AI has progressed from early symbolic systems, such as the **Structure-Mapping Engine** (Falkenhainer et al., 1989), which used hand-crafted representations, to models like the **Latent Relation Mapping Engine** (Turney, 2008), which integrated symbolic rules with statistical learning. The neural era introduced word embeddings for analogy evaluation (Mikolov et al., 2013b), emphasizing local semantic patterns. With LLMs, Webb et al. (2023) demonstrated emergent analogical reasoning, but challenges remain. **AnaloBench** (Ye et al., 2024) shows minimal scaling gains for long-context analogies, while **ANALOGICAL** (Wijesiriwardene et al., 2023) highlights struggles with complex metaphors. Story-level benchmarks like **StoryAnalogy** (Jiayang et al., 2023) and **ARN** (Sourati et al., 2024) reveal difficulties in cross-domain narrative mapping without explicit prompts.

8 Conclusion

This paper systematically dissects the interplay of inductive (System 1) and abductive/deductive (System 2) logical inference within LLMs. We establish that while System 2 pipelines generally yield superior performance—particularly in visual/symbolic modalities and with increasing task difficulty—System 1 remains competitive for textual tasks and, crucially, can outperform System 2 in free-text rule-execution scenarios. These nuanced dynamics extend to broader ICL tasks involving explicit input-output functions. Furthermore, we demonstrate that strategically scaling System 2 through methods like hypothesis selection and iterative refinement largely enhances reasoning capabilities, enabling standard LLMs to approach the performance of specialized reasoning models. Ultimately, this study provides a foundational understanding and actionable guidelines for optimizing LLM reasoning by tailoring logical inference strategies to specific task characteristics.

Limitations

While our extensive experiments and analyses yield rich findings, our exploration is limited to reasoning frameworks for static LLMs. Future research could build on this work by focusing on the tuning stage of LLMs, aiming to develop systems that dynamically balance different types of logical inference. For example, a system capable of automatically identifying the nature of a question and determining whether to apply System 1 or System 2 reasoning could not only maintain or enhance performance but also improve efficiency. Such adaptive reasoning closely mirrors the way humans naturally approach problem-solving.

Ethics Statement

This work aims to advance the understanding of logical inference in LLMs through systematic experimentation and analysis. All LLMs used in this study are publicly available. We strictly prohibit harmful content in the selection, curation, and annotation process of our datasets, ensuring they are free from sensitive or biased material. Our work is conducted with a focus on advancing understanding while adhering to ethical research practices.

Acknowledgements

We thank all the anonymous reviewers and meta reviewers for their valuable comments. The authors of this paper were supported by the ITSP Platform Research Project (ITS/189/23FP) from ITC of Hong Kong, SAR, China, and the AoE (AoE/E-601/24-N), the RIF (R6021-20) and the GRF (16205322) from RGC of Hong Kong, SAR, China. We also thank the support from NVIDIA AI Technology Center (NVAITC).

References

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

Meta AI. 2024. Introducing Llama 3.1: Our most capable models to date.

Jiaxin Bai, Yicheng Wang, Tianshi Zheng, Yue Guo, Xin Liu, and Yangqiu Song. 2024. Advancing abductive reasoning in knowledge graphs through complex logical hypothesis generation. *Preprint*, arXiv:2312.15643.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. *Preprint*, arXiv:1908.05739.

Alexander Bird. 2005. Abductive knowledge and holmesian inference. In Tamar Szabó Gendler and John Hawthorne, editors, *Oxford Studies in Epistemology*, pages 1–31. Oxford University Press.

Yonatan Bitton, Ron Yosef, Eli Strugo, Dafna Shahaf, Roy Schwartz, and Gabriel Stanovsky. 2022. Vasr: Visual analogies of situation recognition. *Preprint*, arXiv:2212.04542.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Chunkit Chan, Xin Liu, Tsz Ho Chan, Jiayang Cheng, Yangqiu Song, Ginny Wong, and Simon See. 2023. Self-consistent narrative prompts on abductive natural language inference. *Preprint*, arXiv:2309.08303.

Jiangjie Chen, Rui Xu, Ziquan Fu, Wei Shi, Zhongqiao Li, Xinbo Zhang, Changzhi Sun, Lei Li, Yanghua Xiao, and Hao Zhou. 2022. E-kar: A benchmark for rationalizing natural language analogical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, page 3941–3955. Association for Computational Linguistics.

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

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *Preprint*, arXiv:2002.05867.

L. Jonathan Cohen. 1982. Intuition, induction, and the middle way. *The Monist*, 65(3):287–301.

I.M. Copi and C. Cohen. 1990. *Introduction to Logic*. Maxwell Macmillan international editions. Macmillan.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2022. Explaining answers with entailment trees. *Preprint*, arXiv:2104.08661.

Google DeepMind. 2024. Google introduces Gemini 2.0: A new AI model for the agentic era.

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

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

Maksym Del and Mark Fishel. 2023. True detective: A deep abductive reasoning benchmark undoable for gpt-3 and challenging for gpt-4. *Preprint*, arXiv:2212.10114.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Li Du, Xiao Ding, Ting Liu, and Bing Qin. 2021. Learning event graph knowledge for abductive reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5181–5190, Online. Association for Computational Linguistics.

Brian Falkenhainer, Kenneth D. Forbus, and Dedre Gentner. 1989. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41(1):1–63.

Wei Fan, Tianshi Zheng, Yiran Hu, Zheye Deng, Weiqi Wang, Baixuan Xu, Chunyang Li, Haoran Li, Weixing Shen, and Yangqiu Song. 2025. Legal rule induction: Towards generalizable principle discovery from analogous judicial precedents. *Preprint*, arXiv:2505.14104.

Peter A Flach and Antonis C Kakas. 2000. *Abduction and Induction: Essays on their Relation and Integration*. Springer Science.

Harry Frankfurt. 1958. Peirce's notion of abduction. *Journal of Philosophy*, 55:593–596.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673.

Dedre Gentner, Keith Holyoak, and Boicho Kokinov. 2001. *The Analogical Mind: Perspectives From Cognitive Science*. MIT Press.

Google. 2024. Introducing Gemini 1.5, google's next-generation AI model.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alex Wardle-Solano, Hannah Szabo, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander R. Fabbri, Wojciech Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. 2024. Folio: Natural language reasoning with first-order logic. *Preprint*, arXiv:2209.00840.

Gilbert H. Harman. 1965. The inference to the best explanation. *The Philosophical Review*, 74(1):88–95.

Kaiyu He, Mian Zhang, Shuo Yan, Peilin Wu, and Zhiyu Zoey Chen. 2024. Idea: Enhancing the rule learning ability of large language model agent through induction, deduction, and abduction. *Preprint*, arXiv:2408.10455.

Carl G. Hempel and Paul Oppenheim. 1948. Studies in the logic of explanation. *Philosophy of Science*, 15(2):135–175.

Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P Burgess, Matko Bosnjak, Murray Shanahan, Matthew Botvinick, Demis Hassabis, and Alexander Lerchner. 2018. Scan: Learning hierarchical compositional visual concepts. *Preprint*, arXiv:1707.03389.

Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. 2022. Stratified rule-aware network for abstract visual reasoning. *Preprint*, arXiv:2002.06838.

Xiaoyang Hu, Shane Storks, Richard Lewis, and Joyce Chai. 2023. In-context analogical reasoning with pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1953–1969, Toronto, Canada. Association for Computational Linguistics.

Cheng Jiayang, Lin Qiu, Tsz Ho Chan, Tianqing Fang, Weiqi Wang, Chunkit Chan, Dongyu Ru, Qipeng Guo, Hongming Zhang, Yangqiu Song, Yue Zhang, and Zheng Zhang. 2023. Storyanalogy: Deriving story-level analogies from large language models to unlock analogical understanding. *Preprint*, arXiv:2310.12874.

Phil Johnson-Laird. 2010. Deductive reasoning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1:8 – 17.

Lara L. Jones, Matthew J. Kmiecik, Jessica L. Irwin, and Robert G. Morrison. 2022. Differential effects of semantic distance, distractor salience, and relations in verbal analogy. *Psychonomic Bulletin & Review*, 29:1480–1491.

Emīls Kadiķis, Vaibhav Srivastav, and Roman Klinger. 2022. Embarrassingly simple performance prediction for abductive natural language inference. *Preprint*, arXiv:2202.10408.

Daniel Kahneman. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York.

Subin Kim, Prin Phunyaphibarn, Donghyun Ahn, and Sundong Kim. 2022. Playgrounds for abstraction and reasoning. In *NeurIPS 2022 Workshop on Neuro Causal and Symbolic AI (nCSI)*.

Chunyang Li, Weiqi Wang, Tianshi Zheng, and Yangqiu Song. 2025. Patterns over principles: The fragility of inductive reasoning in LLMs under noisy observations. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19608–19626, Vienna, Austria. Association for Computational Linguistics.

Peter Lipton. 2000. Inference to the best explanation. In W. Newton-Smith, editor, *A companion to the philosophy of science*, pages 184–193. Blackwell.

Emmy Liu, Graham Neubig, and Jacob Andreas. 2024. An incomplete loop: Instruction inference, instruction following, and in-context learning in language models. *Preprint*, arXiv:2404.03028.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Preprint*, arXiv:1310.4546.

OpenAI. 2024. Hello GPT-4o.

OpenAI. 2024. Introducing openai o1 preview. Accessed: 2025-02-14.

OpenAI. 2025. Openai o3 mini: Pushing the frontier of cost-effective reasoning.

Charles Sanders Peirce. 1958. *Collected Papers of Charles Sanders Peirce*, volume 1-6. Harvard University Press, Cambridge, MA.

Wendy J. Phillips, Janet M. Fletcher, Anthony D. G. Marks, and Donald W. Hine. 2016. Thinking styles and decision making: A meta-analysis. *Psychological Bulletin*, 142(3):260–290.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2021. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. *Preprint*, arXiv:2010.05906.

Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. 2024. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. In *The Twelfth International Conference on Learning Representations*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

J. C. Raven. 1938. *Raven's Progressive Matrices*. Western Psychological Services.

Raymond Reiter. 1987. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Annual Meeting of the Association for Computational Linguistics*.

Joshua S Rule. 2020. *The child as hacker: Building more human-like models of learning*. Ph.D. thesis, MIT.

Merrilee H. Salmon. 1984. *Introduction to Logic and Critical Thinking*. Harcourt Brace Jovanovich, Fort Worth.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using ood examples. *Preprint*, arXiv:2305.15269.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. *Preprint*, arXiv:1409.1556.

Zhivar Sourati, Filip Ilievski, Pia Sommerauer, and Yifan Jiang. 2024. Arn: Analogical reasoning on narratives. *Preprint*, arXiv:2310.00996.

Claire Stevenson, Alexandra Pafford, Han Maas, and Melanie Mitchell. 2024. Can large language models generalize analogy solving like people can?

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2021. Proofwriter: Generating implications, proofs, and abductive statements over natural language. *Preprint*, arXiv:2012.13048.

Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2024. A benchmark for learning to translate a new language from one grammar book. *Preprint*, arXiv:2309.16575.

Paul Thagard. 2024. Can chatgpt make explanatory inferences? benchmarks for abductive reasoning. *Preprint*, arXiv:2404.18982.

P. D. Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

Marianna S. Vendetti, Barbara J. Knowlton, and Keith J. Holyoak. 2012. The impact of semantic distance and induced stress on analogical reasoning: A neurocomputational account. *Cognitive, Affective, & Behavioral Neuroscience*, 12(4):804–812.

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah Goodman. 2024. Hypothesis search: Inductive reasoning with language models. In *The Twelfth International Conference on Learning Representations*.

Taylor Webb, Keith J. Holyoak, and Hongjing Lu. 2023. Emergent analogical reasoning in large language models. *Preprint*, arXiv:2212.09196.

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

Thilini Wijesiriwardene, Ruwan Wickramarachchi, Bimal G. Gajera, Shreeyash Mukul Gowaikar, Chandan Gupta, Aman Chadha, Aishwarya Naresh Reganti, Amit Sheth, and Amitava Das. 2023. Analogical – a novel benchmark for long text analogy evaluation in large language models. *Preprint*, arXiv:2305.05050.

Yudong Xu, Wenhao Li, Pashootan Vaezipoor, Scott Sanner, and Elias B Khalil. 2023. Llms and the abstraction and reasoning corpus: Successes, failures, and the importance of object-based representations. *arXiv preprint arXiv:2305.18354*.

Xiao Ye, Andrew Wang, Jacob Choi, Yining Lu, Shreya Sharma, Lingfeng Shen, Vijay Tiyyala, Nicholas Andrews, and Daniel Khashabi. 2024. Analobench: Benchmarking the identification of abstract and long-context analogies. *Preprint*, arXiv:2402.12370.

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. 2019. Raven: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wenting Zhao, Justin T Chiu, Jena D. Hwang, Faeze Brahman, Jack Hessel, Sanjiban Choudhury, Yejin Choi, Xiang Lorraine Li, and Alane Suhr. 2024. Uncommonsense reasoning: Abductive reasoning about uncommon situations. *Preprint*, arXiv:2311.08469.

Tianshi Zheng, Yixiang Chen, Chengxi Li, Chunyang Li, Qing Zong, Haochen Shi, Baixuan Xu, Yangqiu Song, Ginny Y. Wong, and Simon See. 2025a. The curse of cot: On the limitations of chain-of-thought in in-context learning. *Preprint*, arXiv:2504.05081.

Tianshi Zheng, Jiazheng Wang, Zihao Wang, Jiaxin Bai, Hang Yin, Zheye Deng, Yangqiu Song, and Jianxin Li. 2025b. Enhancing transformers for generalizable first-order logical entailment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5505–5524, Vienna, Austria. Association for Computational Linguistics.

Tianyang Zhong, Yaonai Wei, Li Yang, Zihao Wu, Zhengliang Liu, Xiaozheng Wei, Wenjun Li, Junjie Yao, Chong Ma, Xiang Li, Dajiang Zhu, Xi Jiang, Junwei Han, Dinggang Shen, Tianming Liu, and Tuo Zhang. 2023. Chatabl: Abductive learning via natural language interaction with chatgpt. *Preprint*, arXiv:2304.11107.

Qing Zong, Zhaowei Wang, Tianshi Zheng, Xiyu Ren, and Yangqiu Song. 2024. Comparisonqa: Evaluating factuality robustness of llms through knowledge frequency control and uncertainty. *Preprint*, arXiv:2412.20251.

A Model Details

In our experiments, we tested 15 modern LLM / MLLMs / LRMs with details as follows:

- **Qwen-2.5-7b / Qwen-2.5-72b** (Qwen et al., 2025) is an open-source MoE LLM series, trained with 18 trillion tokens of pre-training corpus and 1 million fine-tuning examples.
- **Llama-3.1-70b / Llama-3.1-405b** (AI, 2024) is an open-source dense LLM series, trained with 15 trillion tokens of pre-training corpus, and adopted DPO (Rafailov et al., 2024) during its alignment stage.
- **GPT-4o-mini / GPT-4o** (OpenAI, 2024) is the latest proprietary LLM series by OpenAI prior to their reasoning models.
- **Gemini-1.5-flash / Gemini-1.5-pro** (Google, 2024) is a proprietary MoE LLM series featuring a long context window of 1 million tokens.
- **Gemini-2.0-flash** (DeepMind, 2024) is the latest Gemini series LLM, offering enhanced multimodal and reasoning performance.
- **Pixtral-12b** (Agrawal et al., 2024) is a lightweight open-source multimodal LLM.
- **Deepseek-V3** (DeepSeek-AI et al., 2024) is the state-of-the-art open-source LLM.
- **Deepseek-R1** (DeepSeek-AI et al., 2025) is the leading open-source LRM trained with reinforcement learning using a rule-based reward system.
- **o1-mini / o1** (OpenAI, 2024) represents the state-of-the-art proprietary LRM series developed by OpenAI.
- **o3-mini** (OpenAI, 2025) is the latest LRM by OpenAI, featured its cost-effectiveness.

The temperature for all LLMs is set to zero in our main experiments, while it is set to 0.4 during the hypothesis sampling in our scaling experiments.

B Difficulty Annotation

The detailed difficulty annotation standards are presented in Table 5. For **EKAR** and **VASR**, we set thresholds for semantic distances to categorize the difficulty into easy, medium, and hard, ensuring comparable sizes across categories.

$$\text{sem\_dist} = \frac{\text{cos\_dist}(A, B) + \text{cos\_dist}(A', B')}{2}$$

For **RAVEN**, we calculate the number of attributes in transition among rows, with fine-grained categorization applied within each question typology. For **List Function**, we use the predefined complexity ranking of mapping functions provided by (Rule, 2020). For **SALT**, we classify the syntax complexity of the translation examples into simple, medium, and complex categories.

Dataset	Determinator	Category	Easy	Medium	Hard
E-KAR	FastText Distance	-	<0.70	0.70∼0.80	>0.80
VASR	VGG Distance	-	<0.70	0.70∼0.76	>0.76
RAVEN	Number of Transitions	center_single	1	2	>=3
		distribute_four	<=2	3	>=4
		distribute_nine	<=2	3	>=4
		in_center_single_out_center_single	<=3	4	>=5
		in_distribute_four_out_center_single	<=3	4	>=5
		up_center_single_down_center_single	<=3	4	>=5
		left_center_single_right_center_single	<=4	5	>=6
List Function	Function Complexity Ranking	-	<=84	85∼170	>=170
SALT	Syntax Complexity	-	simple	intermediate	complex

Table 5: Difficulty classification standards for each datasets in our experiment.

English Sentence	*I like beautiful house.*	*Giant elephant runs quickly.*
Syntax Structure	<pronoun - verb - adjective - noun>	<adjective - noun - verb - adverb>
Grammar Rule	<noun-adjective inversion>	<predicate-subject inversion>
Transition Type	Intra-Constituent	Inter-Constituent
Vocabulary	I → gkt, like → ivo, beautiful → prr, house → cbi	giant → rgd, elephant → krt, runs → uco, quickly → xrk
Translation	*gkt ivo cbi prr.*	*uco xrk rgd krt.*

Table 6: Examples of intra-constituent and inter-constituent syntactic transitions in the SALT dataset.

C Syntax-aware Artificial Language Translation

Syntax-aware Artificial Language Translation (SALT) is a low-resource machine translation (MT) benchmark that we designed and developed to evaluate generalizable in-context learning in large language models. LLMs are required to infer vocabulary mappings as well as syntactic transitions from few-shot demonstrations and apply them to translate a compositionally crafted testing instance. SALT offers two key advantages over other low-resource MT benchmarks: 1) SALT synthesizes out-of-vocabulary strings for the artificial language, **preventing data leakage**, a common issue in other benchmarks. 2) SALT provides **detailed difficulty control** enabled by human-curated syntactic structures with compositional complexities.

The creation of SALT involves two main stages:

1. **Syntax-aware Template Design** In the first stage, we design syntactic rules that involve the permutation or repetition of semantic units in the artificial language, as illustrated in Table 6 and 7. Next, we manually craft templates for few-shot demonstrations with considerations in compositional generalization. We ensure that all the necessary underlying word mappings and syntactic rules required for translating the testing instances can be inferred from the provided few-shot demonstrations.

Difficulty Level	**Syntax Rule**	**Sentence Complexity**
Easy	word-to-word mapping	Simple
Easy	noun repetition	Simple
Easy	noun-adjective inversion	Simple
Easy	predicate-subject inversion	Simple
Medium	word-to-word mapping	Intermediate
Medium	verb repetition	Intermediate
Medium	noun-adjective inversion	Intermediate
Medium	predicate-subject inversion	Intermediate
Hard	word-to-word mapping	Complex
Hard	adjective repetition	Complex
Hard	verb-adverb inversion	Complex
Hard	predicate-subject inversion	Complex

Table 7: Summary of the 12 translation modes in SALT, listing their syntax rule and sentence complexity.

2. **Semantic-aware Data Synthesis** After acquiring the templates, we populate them with semantically appropriate English words using LLM-assisted selection. Next, we randomly assign out-of-vocabulary letter strings as the artificial language equivalents for each English word. Finally, a total of 1,200 questions are sampled—400 at each difficulty level—ensuring comparability in size with other datasets.

D Prompt Templates

Textual Analogy (Induction)

```
Below is an analogy question, where analogy
x:y::x’:y’ exists between the two wordsets, your task
is to finish the second wordset to complete the analogy.

Wordset1: <word_x>:<word_x'>
Wordset2: <word_y>:[missing_word]

Your response should strictly follow the JSON dict format:

{
    "answer": "missing word here"
}
```

Textual Analogy (Automatic)

```
Below is an analogy question, where analogy
x:y::x’:y’ exists between the two wordsets, your task
is to finish the second wordset to complete the analogy.

Wordset1: <word_x>:<word_x'>
Wordset2: <word_y>:[missing_word]

Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "answer": "missing word here"
}
```

Textual Analogy (Abduction)

```
Below is an analogy question, where analogy x:y::x’:y’
exists between the two wordsets, your task is to infer
the relational pattern within wordsets.

Wordset1: <word_x>:<word_x'>
Wordset2: <word_y>:[missing_word]

Your response should strictly follow the JSON dict format:

{
    "reasoning": "reasoning steps here"
    "pattern": "relational pattern here"
}
```

Textual Analogy (Deduction)

```
Below is an analogy question, where analogy x:y::x’:y’
exists between the two wordsets, your task is to
finish the second wordset to complete the analogy.
Here’s the relational pattern: <pattern>

Wordset1: <word_x>:<word_x'>
Wordset2: <word_y>:[missing_word]

Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "answer": "missing word here"
}
```

Visual Analogy (Induction)

```
Below is an analogy question, where analogy x:y::x':y' exists between the two image pairs, your task is to complete the second image pair to complete the analogy.

Image Pair 1: <img_x>:<img_x'>
Image Pair 2: <img_y>:[missing_img]

<Candidate Images>

Your response should strictly follow the JSON dict format:

{
    "answer": "missing image choice here"
}
```

Visual Analogy (Automatic)

```
Below is an analogy question, where analogy x:y::x':y' exists between the two image pairs, your task is to complete the second image pair to complete the analogy.

Image Pair 1: <img_x>:<img_x'>
Image Pair 2: <img_y>:[missing_img]

<Candidate Images>

Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "answer": "missing image choice here"
}
```

Visual Analogy (Abduction)

```
Below is an analogy question, where analogy x:y::x':y' exists between the two image pairs, your task is to infer the relational pattern within image pairs.

Image Pair 1: <img_x>:<img_x'>
Image Pair 2: <img_y>:[missing_img]

<Candidate Images>

Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "pattern": "relational pattern here"
}
```

Visual Analogy (Deduction)

```
Below is an analogy question, where analogy x:y::x':y' exists between the two image pairs, your task is to complete the second image pair to complete the analogy. Here's the relational pattern: <pattern>

Image Pair 1: <img_x>:<img_x'>
Image Pair 2: <img_y>:[missing_img]

<Candidate Images>

Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "answer": "missing image choice here"
}
```

Symbolic Analogy (Induction)

```
Below is a 3x3 matrix of abstracted symbols. The symbols follow a certain rule or pattern in rows. Your task is to infer the missing symbol.

Incomplete Matrix: <incomplete_matrix>

Your response should strictly follow the JSON dict format:

{
    "answer": "missing symbol here"
}
```

Symbolic Analogy (Automatic)

```
Below is a 3x3 matrix of abstracted symbols. The symbols follow a certain rule or pattern in rows. Your task is to infer the missing symbol.

Incomplete Matrix: <incomplete_matrix>

Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "answer": "missing symbol here"
}
```

Symbolic Analogy (Abduction)

```
Below is a 3x3 matrix of abstracted symbols. The symbols follow a certain rule or pattern in rows. Your task is to infer the relational pattern.

Incomplete Matrix: <incomplete_matrix>

Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "pattern": "relational pattern here"
}
```

Symbolic Analogy (Deduction)

```
Below is a 3x3 matrix of abstracted symbols. The symbols follow a certain rule or pattern in rows. Your task is to infer the missing symbol. Here's the relational pattern: <pattern>

Incomplete Matrix: <incomplete_matrix>

Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "answer": "missing symbol here"
}
```

List Function ICL (Induction)

```
Below are several examples of input and output lists.
There exists an unified function that maps the input
list to the output list.

Input 1: <input_list1>, Output 1: <output_list1>
Input 2: <input_list2>, Output 2: <output_list2>
Input 3: <input_list3>, Output 3: <output_list3>

Please infer the output list for the new input list below:
New Input: <new_input_list>

Your response should strictly follow the JSON dict format:

{
    "answer": "output list here"
}
```

List Function ICL (Automatic)

```
Below are several examples of input and output lists.
There exists an unified function that maps the input
list to the output list.

Input 1: <input_list1>, Output 1: <output_list1>
Input 2: <input_list2>, Output 2: <output_list2>
Input 3: <input_list3>, Output 3: <output_list3>

Please infer the output list for the new input list below:
New Input: <new_input_list>

Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "answer": "output list here"
}
```

List Function ICL (Abduction)

```
Below are several examples of input and output lists.
There exists an unified function that maps the input
list to the output list.

Input 1: <input_list1>, Output 1: <output_list1>
Input 2: <input_list2>, Output 2: <output_list2>
Input 3: <input_list3>, Output 3: <output_list3>

Please infer the mapping function in python.
Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "function": "python function here"
}
```

List Function ICL (Deduction)

```
Below are several examples of input and output lists.
There exists an unified function that maps the input
list to the output list. The python code for the
function is: <function>

Input 1: <input_list1>, Output 1: <output_list1>
Input 2: <input_list2>, Output 2: <output_list2>
Input 3: <input_list3>, Output 3: <output_list3>

Please infer the output list for the new input list below:
New Input: <new_input_list>

Your response should strictly follow the JSON dict format:

{
    "reasoning":"reasoning steps here",
    "answer": "output list here"
}
```

SALT ICL (Induction)

```
You are required to translate english sentences to an
artificial language. The translation involves both
vocabulary mapping and syntax rules transition. Below
are translation examples:

English 1: <english_1>, Translation 1: <translation_1>
English 2: <english_2>, Translation 2: <translation_2>
English 3: <english_3>, Translation 3: <translation_3>
English 4: <english_4>, Translation 4: <translation_4>

Please translate this sentence: <english_new>
Your response should strictly follow the JSON dict format:
{
    "translation": "translated sentence here"
}
```

SALT ICL (Automatic)

```
You are required to translate english sentences to an
artificial language. The translation involves both
vocabulary mapping and syntax rules transition. Below
are translation examples:

English 1: <english_1>, Translation 1: <translation_1>
English 2: <english_2>, Translation 2: <translation_2>
English 3: <english_3>, Translation 3: <translation_3>
English 4: <english_4>, Translation 4: <translation_4>

Please translate this sentence: <english_new>
Your response should strictly follow the JSON dict format:
{
    "reasoning":"reasoning steps here",
    "translation": "translated sentence here"
}
```

SALT ICL (Abduction)

```
You are required to study translations from english
sentences to an artificial language. The translation
involves both vocabulary mapping and syntax rules
transition. Below are translation examples:

English 1: <english_1>, Translation 1: <translation_1>
English 2: <english_2>, Translation 2: <translation_2>
English 3: <english_3>, Translation 3: <translation_3>
English 4: <english_4>, Translation 4: <translation_4>

Please infer the word mappings and syntax rules.
Your response should strictly follow the JSON dict format:
{
    "reasoning":"reasoning steps here",
    "vocabulary": "word mappings here",
    "grammar": "syntax rules here"
}
```

SALT ICL (Deduction)

```
You are required to translate english sentences to
an artificial language. The translation involves
both vocabulary mapping and syntax rules transition.
Vocabulary mapping: <vocab>; Syntax rules: <grammar>.
Below are translation examples:

English 1: <english_1>, Translation 1: <translation_1>
English 2: <english_2>, Translation 2: <translation_2>
English 3: <english_3>, Translation 3: <translation_3>
English 4: <english_4>, Translation 4: <translation_4>

Please translate this sentence: <english_new>
Your response should strictly follow the JSON dict format:
{
    "reasoning":"reasoning steps here",
    "translation": "translated sentence here"
}
```

E Interpretation on Task-Format Dependency

We investigated System 2's limitations in free-text generation using the *List Function* dataset, where intermediate rules are evaluatable Python functions. This allows direct assessment of abductive inference accuracy. We compared the accuracy of LLMs generating these Python functions (abduction) against their accuracy in applying ground-truth functions (deduction).

As evidenced by the results in Table 8, the substantially lower abduction accuracy indicates that a primary reason for System 2's failure in free-text rule execution ICL is the insufficient ability of LLMs to precisely generate rules.

Model	Abduction	Deduction
Qwen-2.5-7b	26.80	86.64
Qwen-2.5-72b	50.20	90.72
GPT-4o-mini	40.60	92.56
Average	**39.20**	**89.97**

Table 8: Abduction vs. Deduction Accuracy (%) on List Function Dataset.

Furthermore, to assess the impact of contextual distance from lengthy reasoning chains, characteristic of System 2 and Automatic (CoT) pipelines, we introduced dummy reasoning tokens of varying lengths before the answer in Direct Induction and Automatic pipelines on the List Function dataset (FTG). This simulates how extended context might impair free-text generation precision.

As evidenced by the results in Table 9, performance degrades for both pipelines as token length increases. This suggests that lengthy rationales contribute to task-format sensitivity by disrupting precise free-text output.

Pipeline	Contextual Distance	Qwen-2.5-7b	Qwen-2.5-72b
Direct Induction	0	25.6	47.6
	100	10.4	46.0
	400	9.2	40.4
Automatic (Zero-shot CoT)	0	27.6	46.8
	100	17.6	43.6
	400	16.4	38.8

Table 9: Impact of Dummy Reasoning Tokens on Performance (%) in List Function (FTG).

F Details of Scaling Experiments

This appendix outlines the methodologies for the scaling experiments in Section 6.

- **Figure 4a (Hypothesis Selection):** The LLM first samples multiple candidate hypotheses, ranging from 1 to 10 candidates, using a temperature of 0.4. From these candidates, the LLM then selects the single best hypothesis.
- **Figure 4b (Hypothesis Verification and Refinement):** Initially, a single hypothesis is obtained through the regular abduction process. This hypothesis is then subjected to iterative verification and refinement by the LLM, with this process repeated for multiple rounds.
- **Figure 4c (Combined Selection and Refinement):** This approach begins with the LLM selecting the best hypothesis from several sampled candidates. The chosen hypothesis then undergoes iterative verification and refinement over multiple rounds.
- **Table 4 (Adaptive Scaling for DeepSeek-V3):** This method also combines selection and refinement, but with the LLM autonomously determining the number of refinement rounds within predefined limits. For the *Low Consumption* setting, the LLM selects from 3 candidate hypotheses and refines the chosen one for at most 3 rounds. For the *High Consumption* setting, selection is from 5 candidates, followed by refinement for at most 5 rounds.

G Full Results

The detailed LLM performances in our analogy environement and in-context learning benchmarks is presented in tables below:

- Table 10: Textual Analogy (E-KAR)-MCQ
- Table 11: Visual Analogy (VASR)-MCQ
- Table 12: Symbolic Analogy (RAVEN)-MCQ
- Table 13: Textual Analogy (E-KAR)-FTG
- Table 14: Symbolic Analogy (RAVEN)-FTG
- Table 15: List Function ICL-MCQ
- Table 16: List Function ICL-FTG
- Table 17: SALT ICL-MCQ
- Table 18: SALT ICL-FTG

Model	Pipeline	Easy	Medium	Hard	Total
Qwen-2.5-7b	Induction	65.93	56.32	40.93	52.64
	Automatic	68.45	52.87	39.11	51.36
	Abduction+Deduction	69.40	54.71	44.35	54.33
Qwen-2.5-72b	Induction	76.03	68.74	46.77	61.86
	Automatic	75.39	67.13	49.60	62.26
	Abduction+Deduction	76.97	70.34	51.01	64.34
Llama-3.1-70b	Induction	64.67	56.32	37.90	51.12
	Automatic	73.19	64.14	46.37	59.38
	Abduction+Deduction	73.19	62.53	46.17	58.73
Llama-3.1-405b	Induction	74.76	64.83	43.95	59.05
	Automatic	77.92	68.97	52.62	64.74
	Abduction+Deduction	73.50	67.13	50.60	62.18
GPT-4o-mini	Induction	66.88	54.94	36.49	50.64
	Automatic	63.72	55.40	40.32	51.52
	Abduction+Deduction	63.41	56.78	40.73	52.08
GPT-4o	Induction	73.82	64.83	44.15	58.89
	Automatic	69.72	63.22	48.59	59.05
	Abduction+Deduction	73.50	68.74	51.61	63.14

Table 10: LLM performances on textual analogy dataset (E-KAR) in MCQ task format.

Model	Pipeline	Easy	Medium	Hard	Total
Gemini-1.5-flash	Induction	38.90	30.59	29.38	33.11
	Automatic	54.07	49.83	47.50	50.71
	Abduction+Deduction	59.34	47.73	48.75	51.89
Gemini-1.5-pro	Induction	50.55	45.28	43.13	46.55
	Automatic	65.49	54.37	59.06	59.24
	Abduction+Deduction	65.71	57.34	59.38	60.65
Gemini-2.0-flash	Induction	52.31	47.38	47.50	49.07
	Automatic	63.96	60.84	61.56	62.06
	Abduction+Deduction	67.47	59.44	65.62	63.62
Pixtral-12b	Induction	32.53	24.30	17.81	25.54
	Automatic	33.85	32.87	30.94	32.74
	Abduction+Deduction	41.54	39.34	35.31	39.12
GPT-4o-mini	Induction	34.73	26.57	25.31	29.03
	Automatic	51.43	41.61	40.00	44.54
	Abduction+Deduction	51.21	41.43	44.06	45.36
GPT-4o	Induction	54.95	47.90	46.56	49.96
	Automatic	66.37	55.07	59.06	59.84
	Abduction+Deduction	68.13	59.97	60.94	62.95

Table 11: LLM performances on visual analogy dataset (VASR) in MCQ task format.

Model	Pipeline	Easy	Medium	Hard	Total
Qwen-2.5-7b	Induction	29.10	19.26	11.39	19.94
	Automatic	29.10	20.35	14.43	21.29
	Abduction+Deduction	30.10	21.43	16.46	22.64
Qwen-2.5-72b	Induction	40.55	28.57	18.99	29.39
	Automatic	51.24	38.74	26.08	38.76
	Abduction+Deduction	54.48	43.72	36.20	44.80
Llama-3.1-70b	Induction	38.06	28.35	18.73	28.44
	Automatic	52.99	36.58	28.35	39.24
	Abduction+Deduction	49.50	36.36	34.43	39.95
Llama-3.1-405b	Induction	54.23	38.10	25.06	39.16
	Automatic	53.48	38.53	28.35	40.11
	Abduction+Deduction	64.93	47.62	37.72	50.04
GPT-4o-mini	Induction	36.82	22.51	15.44	24.86
	Automatic	37.56	26.41	12.91	25.73
	Abduction+Deduction	36.32	25.11	22.78	27.96
GPT-4o	Induction	41.79	29.87	17.22	29.71
	Automatic	58.21	41.13	35.44	44.80
	Abduction+Deduction	55.47	35.93	31.39	40.75

Table 12: LLM performances on symbolic analogy dataset (RAVEN) in MCQ task format.

Model	Pipeline	Easy	Medium	Hard	Total
Qwen-2.5-7b	Induction	28.08	22.99	16.33	21.63
	Automatic	28.71	25.98	19.56	24.12
	Abduction+Deduction	27.76	23.68	17.74	22.36
Qwen-2.5-72b	Induction	35.02	29.20	21.77	27.72
	Automatic	33.75	28.74	22.18	27.40
	Abduction+Deduction	34.07	29.66	21.98	27.72
Llama-3.1-70b	Induction	29.02	22.07	15.32	21.15
	Automatic	32.81	23.45	19.15	24.12
	Abduction+Deduction	29.97	25.52	18.95	24.04
Llama-3.1-405b	Induction	28.71	24.60	16.94	22.60
	Automatic	29.34	25.75	18.75	23.88
	Abduction+Deduction	32.18	27.59	19.76	25.64
GPT-4o-mini	Induction	28.08	22.99	16.33	21.63
	Automatic	29.34	25.29	20.16	24.28
	Abduction+Deduction	29.97	25.75	19.15	24.20
GPT-4o	Induction	32.81	26.90	19.35	25.40
	Automatic	32.18	26.21	20.77	25.56
	Abduction+Deduction	31.23	27.59	20.36	25.64

Table 13: LLM performances on textual analogy dataset (E-KAR) in FTG task format.

Model	Pipeline	Easy	Medium	Hard	Total
Qwen-2.5-7b	Induction	19.15	8.87	5.57	11.12
	Automatic	0.75	0.87	0.00	0.56
	Abduction+Deduction	1.00	2.60	1.77	1.83
Qwen-2.5-72b	Induction	37.81	20.13	13.42	23.67
	Automatic	17.91	5.41	1.52	8.18
	Abduction+Deduction	25.37	13.85	8.86	15.97
Llama-3.1-70b	Induction	30.35	13.20	8.10	17.08
	Automatic	18.16	7.14	4.81	9.93
	Abduction+Deduction	9.45	7.58	6.08	7.70
Llama-3.1-405b	Induction	42.29	20.78	13.42	25.34
	Automatic	30.85	12.99	7.34	16.92
	Abduction+Deduction	28.61	16.45	12.15	18.98
GPT-4o-mini	Induction	26.37	12.34	8.61	15.65
	Automatic	11.19	4.76	2.53	6.12
	Abduction+Deduction	11.69	6.93	3.54	7.39
GPT-4o	Induction	37.81	18.40	10.89	22.24
	Automatic	16.17	9.09	5.82	10.33
	Abduction+Deduction	25.12	14.07	9.37	16.12

Table 14: LLM performances on symbolic analogy dataset (RAVEN) in FTG task format.

Model	Pipeline	Easy	Medium	Hard	Total
Qwen-2.5-7b	Induction	47.69	33.57	29.87	37.28
	Automatic	60.42	44.21	39.24	48.24
	Abduction+Deduction	64.81	32.97	38.23	49.36
Qwen-2.5-72b	Induction	65.05	46.34	40.51	50.96
	Automatic	69.44	52.25	44.81	55.84
	Abduction+Deduction	68.52	49.41	45.57	54.80
GPT-4o-mini	Induction	59.03	45.86	33.92	46.64
	Automatic	61.81	53.90	42.28	52.96
	Abduction+Deduction	66.20	52.01	44.80	54.64

Table 15: LLM performances on List Function dataset in MCQ task format.

Model	Pipeline	Easy	Medium	Hard	Total
Qwen-2.5-7b	Induction	65.18	36.24	9.75	37.60
	Automatic	54.59	24.71	7.50	29.36
	Abduction+Deduction	57.88	24.71	8.00	30.64
Qwen-2.5-72b	Induction	79.06	49.65	17.25	49.28
	Automatic	74.59	44.94	13.75	45.04
	Abduction+Deduction	80.47	47.53	17.75	48.32
GPT-4o-mini	Induction	75.53	43.53	13.75	44.88
	Automatic	65.65	32.94	10.50	36.88
	Abduction+Deduction	73.41	41.65	14.00	43.60

Table 16: LLM performances on List Function dataset in FTG task format.

Model	Pipeline	Easy	Medium	Hard	Total
Qwen-2.5-7b	Induction	21.50	16.75	10.00	16.08
	Automatic	36.00	31.75	19.75	29.17
	Abduction+Deduction	35.25	31.50	22.75	29.83
Qwen-2.5-72b	Induction	58.50	54.25	52.00	54.92
	Automatic	61.25	60.00	60.00	60.42
	Abduction+Deduction	60.75	62.00	63.25	62.00
GPT-4o-mini	Induction	63.50	56.75	39.75	53.33
	Automatic	53.00	47.25	43.50	47.92
	Abduction+Deduction	64.75	61.25	53.25	59.75

Table 17: LLM performances on SALT dataset in MCQ task format.

Model	Pipeline	Easy	Medium	Hard	Total
Qwen-2.5-7b	Induction	37.50	15.25	2.00	18.25
	Automatic	29.00	17.25	6.75	17.67
	Abduction+Deduction	29.25	16.00	6.50	17.25
Qwen-2.5-72b	Induction	51.25	33.50	19.50	34.75
	Automatic	38.00	35.25	26.75	33.33
	Abduction+Deduction	33.50	30.25	30.00	31.25
GPT-4o-mini	Induction	66.25	25.00	17.25	36.17
	Automatic	34.00	25.50	20.00	26.50
	Abduction+Deduction	38.75	34.00	25.75	32.83

Table 18: LLM performances on SALT dataset in FTG task format.