



HiBench: Benchmarking LLMs Capability on Hierarchical Structure Reasoning

Zhuohang Jiang*
The Hong Kong Polytechnic
University
Hong Kong SAR, China
zhuohang.jiang@connect.polyu.hk

Pangjing Wu*
The Hong Kong Polytechnic
University
Hong Kong SAR, China
pang-jing.wu@connect.polyu.hk

Ziran Liang*
The Hong Kong Polytechnic
University
Hong Kong SAR, China
ziran.liang@connect.polyu.hk

Peter Q. Chen*
The Hong Kong Polytechnic
University
Hong Kong SAR, China
peter-q.chen@connect.polyu.hk

Xu Yuan*
The Hong Kong Polytechnic
University
Hong Kong SAR, China
xander.yuan@connect.polyu.hk

Ye Jia*
The Hong Kong Polytechnic
University
Hong Kong SAR, China
ye-aimmeng.jia@connect.polyu.hk

Jiancheng Tu*
The Hong Kong Polytechnic
University
Hong Kong SAR, China
jiancheng.tu@connect.polyu.hk

Chen Li
The Hong Kong Polytechnic
University
Hong Kong SAR, China
richard-chen.li@polyu.edu.hk

Peter H.F. Ng
The Hong Kong Polytechnic
University
Hong Kong SAR, China
peter.nhf@polyu.edu.hk

Qing Li†
The Hong Kong Polytechnic
University
Hong Kong SAR, China
csqli@comp.polyu.edu.hk

Abstract

Structure reasoning is a fundamental capability of large language models (LLMs), enabling them to reason about structured common-sense and answer multi-hop questions. However, existing benchmarks for structure reasoning mainly focus on horizontal and coordinate structures (e.g. graphs), overlooking the hierarchical relationships within them. Hierarchical structure reasoning is crucial for human cognition, particularly in memory organization and problem-solving. It also plays a key role in various real-world tasks, such as information extraction and decision-making. To address this gap, we propose HiBench, the first framework designed to systematically benchmark the hierarchical reasoning capabilities of LLMs from initial structure generation to final proficiency assessment. It encompasses six representative scenarios, covering both fundamental and practical aspects, and consists of 30 tasks with varying hierarchical complexity, totaling 39,519 queries. To evaluate LLMs comprehensively, we develop five capability dimensions that depict

different facets of hierarchical structure understanding. Through extensive evaluation of 20 LLMs from 10 model families, we reveal key insights into their capabilities and limitations: 1) existing LLMs show proficiency in basic hierarchical reasoning tasks; 2) they still struggle with more complex structures and implicit hierarchical representations, especially in structural modification and textual reasoning. Based on these findings, we create a small yet well-designed instruction dataset, which enhances LLMs' performance on HiBench by an average of 88.84% (Llama-3.1-8B) and 31.38% (Qwen2.5-7B) across all tasks. The HiBench dataset and toolkit are available at <https://github.com/jzzzzh/HiBench> to encourage evaluation.

CCS Concepts

• Computing methodologies → Natural language processing.

Keywords

Hierarchical Reasoning, Benchmark, Natural Language Processing, Large Language Models

ACM Reference Format:

Zhuohang Jiang, Pangjing Wu, Ziran Liang, Peter Q. Chen, Xu Yuan, Ye Jia, Jiancheng Tu, Chen Li, Peter H.F. Ng, and Qing Li. 2025. HiBench: Benchmarking LLMs Capability on Hierarchical Structure Reasoning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25)*, August 3–7, 2025, Toronto, ON, Canada. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3711896.3737378>

*Authors contributed equally to this research.

†Corresponding Author.



This work is licensed under a Creative Commons Attribution 4.0 International License. *KDD '25, Toronto, ON, Canada*

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1454-2/2025/08

<https://doi.org/10.1145/3711896.3737378>

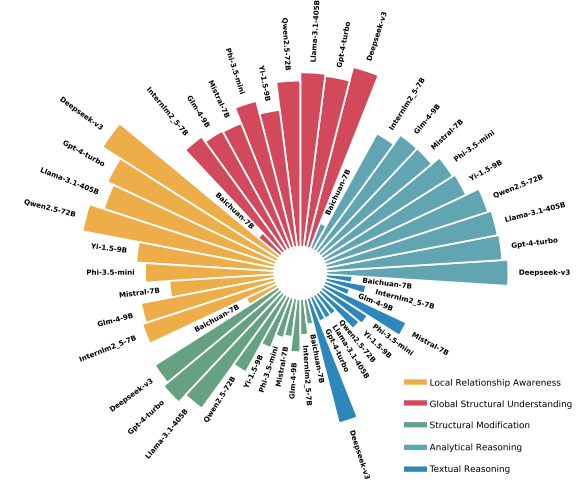
1 Introduction

Recently, Large Language Models (LLMs) have shown remarkable performance across a variety of tasks, such as conversational Artificial Intelligence (AI) [4, 7, 40], text summarization [27, 68], language translation [23, 29, 36], programming assistance [35, 45]. These advancements have driven substantial progress in practical applications, including healthcare [24, 42, 50], finance [60, 66], and software development [33, 39, 59]. Notably, the emergent cognitive capabilities in LLMs have been observed to increasingly mirror certain aspects of human intelligence, which suggests potential parallels between LLMs' behavior and human cognitive processes, arousing discussions about possible pathways to Artificial General Intelligence (AGI) [19, 71]. One fundamental principle of human cognition is hierarchical reasoning, which is essential for memory organization, problem-solving, and decision-making [20, 43, 49], allowing humans to understand and organize complex relationships with structured knowledge effectively. Consequently, evaluating whether and to what extent LLMs exhibit hierarchical reasoning is crucial for further investigating the alignment between their capabilities and human cognition.

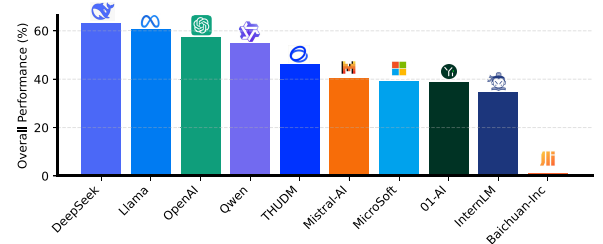
While numerous benchmarks have been developed to assess various cognitive capabilities of LLMs, such as memory retrieval [28], commonsense understanding [30, 47], and structure reasoning [44, 72], there is a significant gap when it comes to hierarchical reasoning evaluation. Existing structure reasoning benchmarks for LLMs primarily focus on tasks involving horizontal or coordinate structures, such as graphs [12, 13, 55] or tables [52, 57], while overlooking the critical hierarchical nature of cognitive reasoning, where information is processed across different levels of abstraction. This absence hinders a comprehensive understanding of LLMs' actual cognitive capabilities. To fill the essential gap in current evaluation systems for LLMs, we propose **HiBench**, the first benchmark specifically designed to evaluate LLMs' hierarchical reasoning capabilities, providing a systematic framework for assessing how LLMs organize, process, and reason with multi-level information across multiple capability dimensions and scenarios.

As diverse task demands drive the evolution of cognitive competencies, HiBench introduces 30 carefully crafted tasks comprising 39,519 queries to evaluate the hierarchical reasoning capabilities of LLMs. These tasks are systematically organized into six distinct scenarios, covering both fundamental and practical aspects of hierarchical reasoning. Specifically, the fundamental aspect comprises three scenarios: *Binary Tree*, *Multiple Tree*, and *JSON*, accounting for 22 tasks. These scenarios are designed to assess LLMs' behavior in processing and manipulating abstract hierarchical structures, which reflect basic yet essential hierarchical reasoning capabilities. The practical aspect features three real-world scenarios: *Code*, *Formula*, and *Paper*, consisting of eight specific tasks. These scenarios incorporate hierarchical reasoning into complex application contexts, enabling the evaluation of how well LLMs handle hierarchical information, providing a robust measure of advanced hierarchical reasoning proficiency.

To provide a comprehensive and thorough assessment of LLMs' hierarchical reasoning capabilities, HiBench establishes a multi-dimensional evaluation system comprising five essential dimensions: *Relationship Awareness*, *Structural Understanding*, *Structural*



(a) Performance of each Family's State-of-the-art LLMs over Five Dimensions of Hierarchical Reasoning Capabilities.



(b) Overall Performance of LLM Families on HiBench.

Figure 1: Performance Distribution of LLM Model Families on HiBench.

Manipulation, *Analytical Reasoning*, and *Textual Reasoning*. These dimensions capture distinct yet progressive aspects of hierarchical reasoning, from recognizing and navigating structures to performing complex reasoning tasks. Furthermore, HiBench incorporates comparative experiments across varying *structure complexity*, *contextual learning paradigms*, and *structure representations*, enabling the investigation of critical factors affecting LLMs' performance across the five evaluation dimensions. As summarized in Figure 1, extensive experimental results demonstrate that current popular LLMs from various families possess preliminary hierarchical reasoning capabilities with general performance over 40.0% accuracy. Our findings reveal several key insights: 1) LLMs generally demonstrate more substantial capabilities in relationship awareness, structural understanding, and analytical reasoning than structural manipulation and textual reasoning; 2) As the complexity of the hierarchical structure increases, whether in depth or breadth, the challenge for LLMs also intensifies; 3) Structure representation with explicit hierarchical information enhances LLMs' reasoning capabilities; 4) LLMs perform more effectively when contextual semantics align with real-world hierarchical relationships; 5) Contextual learning strategies can enhance LLMs' performance, while the benefits of simple Chain-of-Thought (CoT) prompting are limited.

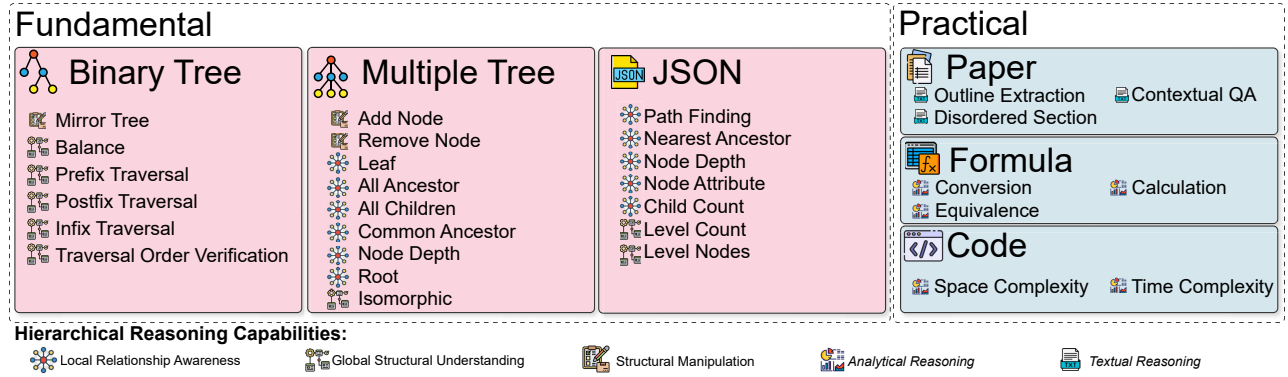


Figure 2: Comprehensive Breakdown of Hierarchical Reasoning Scenarios and Tasks in HiBench.

As LLMs still show room for improvement in hierarchical structure reasoning, we construct a carefully designed instruction dataset consisting of 14,623 question-answer pairs across six scenarios, guided by insights from our findings. The dataset targets LLMs' weaknesses in hierarchical reasoning, focusing on complex structures, implicit representations, and counterfactual configurations. After instruction finetuning, two small-scale LLMs, Llama-3.1-8B [18] and Qwen2.5-7B [63], achieved higher performance on HiBench by 88.84% (Llama-3.1-8B) and 31.38% (Qwen2.5-7B) across all tasks, compared to their vanilla versions. They even exceed large-scale models like Llama-3.1-405B by up to 7.31% (Llama-3.1-8B) and 18.06% (Qwen2.5-7B), and the GPT-4 [2] by up to 6.53% (Qwen2.5-7B) and 0.2% (Llama-3.1-8B), which indicates that a small-scale high-quality dataset can inspire LLMs' hierarchical reasoning capabilities. However, the performance on some tasks remains far less than average, making it an open question to enhance LLMs' hierarchical reasoning capabilities.

Our contributions are summarized as follows:

- We propose HiBench, the first benchmark specifically designed to comprehensively evaluate the hierarchical reasoning capabilities of LLMs.
- We conduct extensive experiments on 20 LLMs across 10 well-known families, uncovering both the strengths and limitations of LLMs in hierarchical reasoning and providing new insights for further advancements.
- By constructing an instruction dataset that targets LLMs' weaknesses in hierarchical reasoning and finetuning small-scale LLMs, we enhance their effectiveness in hierarchical reasoning tasks, outperforming state-of-the-art GPT-4 by 6.53% at most.

2 Related Works

2.1 LLMs on Structure Reasoning

With the growing popularity of LLMs, researchers have begun to deeply explore the combination of LLMs and structured data, such as graphs. Early studies primarily focused on empirical performance evaluations. For instance, GraphBERT [69], GraphTransformer [67], and GraphT5 [31] investigated whether LLMs can comprehend structured graph data, laying a solid foundation for applying LLMs

to graph-related tasks. In addition, Luo *et al.* [38] systematically evaluated LLMs' graph reasoning capabilities through GraphInstruct, while Dai *et al.* [13] highlighted the substantial gap in LLMs' understanding of graph structures. However, most of these studies have been limited to horizontal graph reasoning tasks, ignoring the crucial aspect of hierarchical structure reasoning. This limitation motivates us to investigate the capabilities of LLMs in handling hierarchical architectures, which are more typically required in real-world applications.

2.2 Implicit Hierarchical Thinking in LLMs

The investigation of implicit hierarchical thinking in LLMs has emerged as a prominent research focus in recent years. Some studies have shown that LLMs do not always rely on explicit, step-by-step reasoning but may instead perform implicit reasoning through their internal hierarchical structures. For example, Deng *et al.* [15] proposed a method for transitioning from explicit CoT reasoning to implicit CoT via knowledge distillation, enhancing LLMs' reasoning capabilities. In the context of task planning, LLMs have demonstrated the capability to tackle complex problems through hierarchical decomposition. Yao *et al.* [64], introduced the Tree of Thoughts framework to optimize LLMs' problem-solving capability through hierarchical structures. Similarly, He *et al.* [25] proposed a dual-process framework for dialogue planning that mimics human hierarchical thinking in planning tasks. While significant progress has been made in understanding the implicit hierarchical thinking of LLMs across various domains, a key challenge remains: how to efficiently harness these capabilities to enhance their performance on complex, real-world tasks.

3 Task Taxonomy

This section introduces the task taxonomy of HiBench, which encompasses a wide range of tasks designed to assess hierarchical reasoning capabilities of LLMs, as shown in Figure 2. Recognizing that different task scenarios impose varying demands on hierarchical reasoning, the taxonomy classifies these tasks based on their structural complexity and cognitive requirements. By systematically organizing both fundamental and practical tasks, this taxonomy underscores the pivotal role of hierarchical reasoning in LLM performance. Furthermore, it facilitates a more fine-grained and

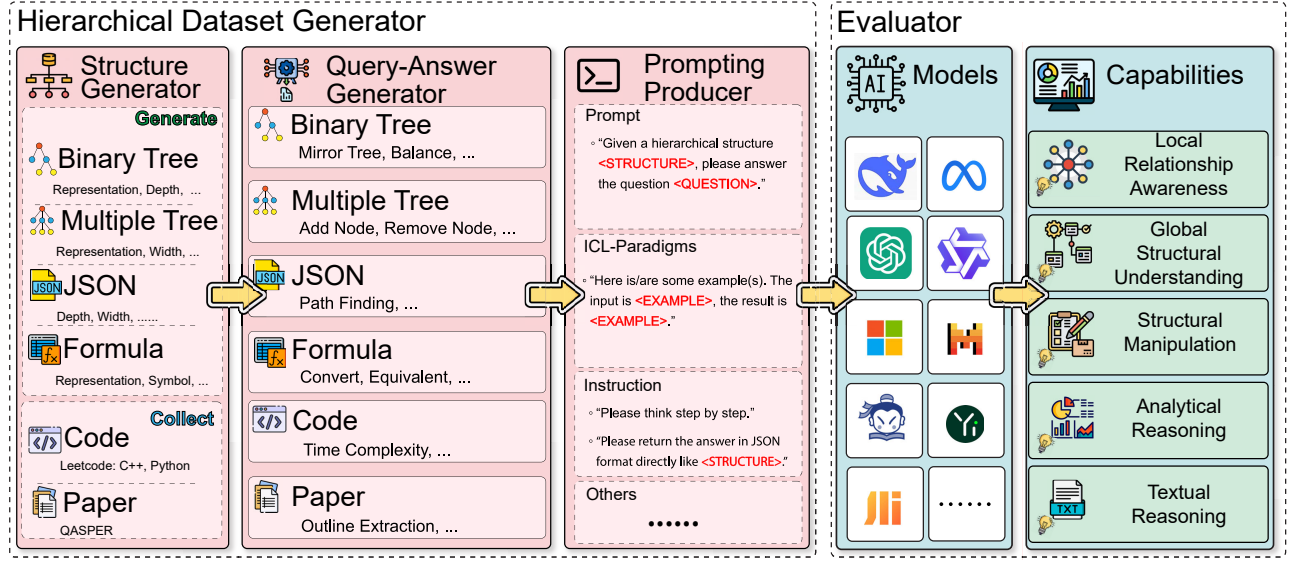


Figure 3: Overview of the HiBench Framework, Encompassing Pipelines from Hierarchical Dataset Generator to Evaluator.

context-specific evaluation of LLM capabilities, providing actionable insights for future model development and refinement.

3.1 Fundamental Aspect

Fundamental tasks are designed to assess LLMs' capabilities to understand and manipulate hierarchical structure data, such as tree-based data and semi-structured formats like JSON. These tasks evaluate whether LLMs can effectively perform hierarchical reasoning across a range of hierarchical representations.

Scenario 1: Binary Tree. The binary tree is a fundamental data structure widely used in computer science. Using a generalized random structure generator, we construct binary trees with varying depths to introduce different levels of complexity, then design six hierarchical tasks upon them, including *Balance*, *Prefix Traversal*, *Infix Traversal*, *Postfix Traversal*, *Traversal Order Verification* and *Mirror Tree*, to evaluate LLMs' hierarchical understanding of binary tree structures.

Scenario 2: Multiple Trees. Multiple trees are widely utilized in diverse domains such as database indexing, file systems, and network architectures, where they excel in representing complex hierarchical arrangements beyond the capabilities of binary trees. To comprehensively assess LLMs' reasoning capabilities in these more intricate settings, we employ a randomized tree generator to construct multiple trees with varying levels of complexity in both breadth and depth. Based on these structures, we develop nine hierarchical tasks: *Add Node*, *All Ancestor*, *All Children*, *Common Ancestor*, *Isomorphic*, *Remove Node*, *Node Depth*, *Leaf* and *Root*.

Scenario 3: JSON. JSON is a semi-structured data interchange format capable of representing hierarchical structures through nested objects and arrays. As a standardized and widely supported format, it is well-suited for evaluating structured data understanding. To this end, we randomly generate JSON files with different

breadth and depth and design the following seven tasks: *Child Count*, *Node Depth*, *Level Count*, *Node Attribute*, *Level Nodes*, *Path Finding*, and *Nearest Ancestor*. These tasks examine LLMs' capability to parse JSON data, comprehend hierarchical relationships, and analyze paths across multiple dimensions.

3.2 Practical Aspect

In contrast to the fundamental aspect, the practical aspect emphasizes real-world applications that vary in information format and structural complexity, such as formulas, code, and academic papers. These tasks assess the versatility and capability of LLMs to apply hierarchical reasoning in tackling noisy, diverse, and complex real-world challenges.

Scenario 4: Formula. Mathematical formulas inherently exhibit certain hierarchical information in prefix, infix, and postfix notations. Therefore, we randomly generate a series of formulas with varying levels of complexity, controlled by formula length, numerical magnitude, and symbolic complexity, and construct three types of formula comprehension tasks: *Conversion*, *Calculation* and *Equivalence*, to evaluate LLMs' hierarchical understanding of mathematical expressions.

Scenario 5: Code. Given the hierarchical nature of code structure and logic, we collect diverse C++ and Python code samples from Github¹ and LeetCode² platforms, and construct two types of tasks, *Space Complexity* and *Time Complexity*, aiming at evaluating the capability of LLMs to reason about code efficiency and structural complexity.

Scenario 6: Paper. Since academic papers are hierarchically structured textual documents, we curate a subset of papers from the QASPER [14] benchmark to evaluate LLMs' ability to reason

¹<https://github.com>

²<https://leetcode.com>

about hierarchical textual information through three task types: *Contextual QA*, *Disordered Section*, and *Outline Extraction*.

4 The HiBench

In this section, we introduce the overall architecture of HiBench, a comprehensive and systematic benchmark developed to assess the hierarchical reasoning capabilities of LLMs. It facilitates user adoption and practical use by offering a well-established and streamlined workflow. The architecture comprises two main components: the *Hierarchical Dataset Constructor* and the *Evaluator*. The *Hierarchical Dataset Constructor* systematically generates benchmark samples with varying complexity, while the *Evaluator* quantifies model performance across five refined capability dimensions. An overview of the HiBench architecture is presented in Figure 3.

4.1 Hierarchical Dataset Generator

The data generation process of HiBench consists of three main stages: *Structure Generator*, *Query-Answer Generator*, and *Prompting Producer*. Building upon the task taxonomy mentioned in Section 3, the complexity of specific tasks within each scenario varies due to the intricacy of their hierarchical structures. To ensure a thorough evaluation, *Structure Generator* constructs various hierarchical structures tailored to each scenario for subsequent processes. Next, *Query-Answer Generator* leverages these pre-constructed structures to produce corresponding queries for sub-tasks in different scenarios. Finally, *Prompting Producer* transforms these queries into well-formatted prompts, adapting them for LLM input while ensuring consistency across evaluations.

Structure Generator. Given that the task taxonomy of HiBench spans diverse application scenarios, each with specialized requirements and varying levels of complexity, tailoring structure generators to each scenario is necessary. For the four scenarios, Binary Tree, Multiple Tree, JSON, and Formula, dedicated structure generators autonomously and efficiently construct large-scale, diverse instances, providing rich and challenging hierarchical structures for comprehensive evaluation. Specifically, the tree generator constructs hierarchical tree data by varying key parameters such as out-degree, depth, and node number. The JSON generator creates data by adjusting two key dimensions: width and depth, while the formula generator adjusts formula length, numerical complexity, and symbolic complexity to introduce varying levels of difficulty. In contrast, for the remaining two scenarios, Code and Paper, their structure generators rely on collecting hierarchically structured data from existing real-world sources. Specifically, the code generator gathers C++ and Python samples from GitHub and LeetCode, while the paper generator reorganizes academic texts sourced from the QASPER dataset. These data capture the inherent hierarchical structures of code and academic writing, ensuring both diversity and real-world representativeness.

Query-Answer Generator. Since each task scenario consists of multiple sub-tasks, we generate corresponding query-answer pairs based on the hierarchical structures constructed in the previous stage. A variety of algorithms are employed to tailor queries to specific sub-tasks. For instance, the prefix traversal algorithm generates query-answer pairs for the corresponding sub-task using

Table 1: Basic Statistics of HiBench.

Scenarios	Tasks	Sub-Tasks	Queries	Avg. Length
Binary Tree	6	216	4,968	2,824.3
Multiple Tree	9	162	9,558	915.0
JSON	7	300	2,586	1,349.8
Formula	3	1,458	18,954	516.8
Code	2	12	1,200	1,296.9
Paper	3	9	2,253	26,759.2
Total (HiBench)	30	2,157	39,519	1,725.9

the binary tree structure. In scenarios such as academic papers, where fixed answers are not always available, we rely on manual annotations to produce accurate query-answer pairs.

Prompting Producer. To support comprehensive evaluation across diverse reasoning types and task formats, we incorporate well-known prompt engineering techniques like In-context Learning (ICL) [16] and CoT [70] into the query generation. This design allows us to assess the impact of prompt engineering on LLM performance. Additionally, we provide explicit output format instructions to ensure that responses adhere to predefined structural and stylistic constraints, thereby facilitating consistent and reliable evaluation.

Table 1 presents a statistical overview of HiBench, comprising 30 tasks and 39,519 queries across six hierarchical scenarios. It serves as a comprehensive benchmark for evaluating the hierarchical reasoning capabilities of LLMs, featuring diverse query lengths and a broad spectrum of sub-tasks.







4.2 Evaluator

In human cognition progress, hierarchical reasoning is represented from understanding local relationships to grasping global structures, performing structural manipulations, and reasoning within increasingly complex information contexts, reflecting a growing cognitive capability. Building upon this, we define five dimensions to evaluate LLMs' hierarchical reasoning capabilities, each mirroring a stage of human hierarchical reasoning:

- *Local Relationship Awareness* assesses the capability to recognize immediate connections, such as parent-child relationships in trees or code dependencies.
- *Global Structural Understanding* involves grasping the overall organization and coherence of hierarchical structures.
- *Structural Manipulation* evaluates the capability to modify structures, such as code refactoring or tree transformations.
- *Analytical Reasoning* measures the capability to derive insights through logical inference and quantitative analysis.
- *Textual Reasoning* examines the ability to comprehend hierarchical structures embedded within complex, context-rich textual information.

Based on the characteristics of each task, we assign it to one of the five evaluation dimensions, as depicted in Figure 2. This categorization reflects the underlying cognitive and computational demands of hierarchical structures, offering a unified schema for systematically assessing the hierarchical reasoning abilities of LLMs.

Table 2: HiBench Leaderboard: Categorizing Models by Open-Source Status and Family.*

Model Family	Model Name	Fundamental Aspect			Practical Aspect			Average	Rank
		 Binary	 Multiple	 JSON	 Code	 Formula	 Paper		
Closed-Source									
OpenAI	GPT-3.5 [46]	39.19	54.22	53.49	66.50	54.54	-**	53.59	6
	GPT-4 [2]	56.29	73.64	63.59	70.75	54.37	38.35	59.50	2
Open-Source									
01-AI	Yi-1.5-9B [65]	26.18	47.67	54.23	66.75	42.42	10.27	41.25	9
Qwen	Qwen2.5-0.5B [63]	1.99	15.03	15.31	2.50	8.83	3.98	7.94	19
	Qwen2.5-1.5B [63]	19.43	43.30	35.47	56.75	21.17	25.54	32.02	17
	Qwen2.5-3B [63]	24.32	48.95	39.71	60.25	39.78	32.38	40.52	10
	Qwen2.5-7B [63]	33.11	59.01	48.58	69.25	41.27	42.08	49.93	7
	QwQ-32B [53]	42.91	74.12	73.80	13.75	16.82	17.22	39.77	12
	Qwen2.5-72B [63]	45.61	63.08	70.50	69.25	51.75	39.72	56.65	3
Baichuan Inc.	Baichuan-7B [61]	2.62	1.75	4.33	0.25	0.93	0.00	1.65	20
DeepSeek	DeepSeek-V3 [34]	70.23	72.59	77.65	72.50	57.40	42.99	65.56	1
Meta	Llama-3.2-1B [18]	9.57	20.45	23.18	44.25	17.02	13.30	21.30	18
	Llama-3.2-3B [18]	31.79	34.70	29.31	60.50	23.20	28.07	34.59	15
	Llama-3.1-8B [18]	14.74	28.76	49.29	61.75	26.35	16.23	32.85	16
	Llama-3.1-70B [18]	54.73	65.04	63.03	68.75	42.14	43.71	56.24	4
	Llama-3.1-405B [18]	64.49	59.95	55.61	53.37	63.45	16.46	55.53	5
Microsoft	Phi-3.5-mini-3.8B [1]	27.36	38.24	57.52	65.25	41.76	11.51	40.27	11
Mistral	Mistral-7B [26]	26.14	31.00	49.19	58.75	40.08	27.10	38.71	13
SHAILab	InternLM2.5-7B [6]	34.28	39.02	54.62	65.75	23.27	6.24	37.20	14
THUDM	GLM-4-9B [22]	36.72	46.26	48.60	66.50	32.11	37.60	44.63	8
Average Performance		33.08	44.61	49.94	55.53	34.43	23.83	40.48	-

* The best results for each task are marked in **bold**, and the second-best results are marked with underline.

** Queries exceed the maximum context token limitation of GPT-3.5.

5 Experiments

In this section, we present the experimental results evaluating the performance of LLMs on hierarchical reasoning using our HiBench.

5.1 Experimental Setup

Models. In the HiBench, we evaluate 20 LLMs from 10 model families. These LLMs are categorized into four main groups, including the *GPT family*, the *Llama family*, the *Qwen family*, and *Other open-source models*.

- **GPT Family:** The GPT family [2, 46] includes a series of advanced LLMs developed by OpenAI and known for their superior natural language processing capabilities. In our HiBench, we choose GPT-3.5 and GPT-4 for evaluation to fully demonstrate their hierarchical reasoning performance.
- **Llama Family:** The Llama family [18] is a collection of open-source LLMs from Meta, noted for their excellent multilingual support, extended context processing capabilities, and optimized architectural design. We employ five Llama models in HiBench, including Llama-3.2-1B, Llama-3.2-3B, Llama-3.1-8B, Llama-3.1-70B, and Llama-3.1-405B.

- **Qwen Family:** The Qwen family [63] comprises a set of open-source LLMs developed by the Alibaba DAMO Academy, known for their strong instruction-following capabilities, extended long-context handling, and improved understanding of structured data. In our setting, we select the following models to benchmark their hierarchical reasoning behaviors: Qwen2.5-0.5B, Qwen2.5-1.5B, Qwen2.5-3B, Qwen2.5-7B, QwQ-32B, and Qwen2.5-72B.
- **Other Open-source LLMs:** Our benchmark also covers representative LLMs from seven other well-known model families, including ChatGLM [22], Phi [1], InternLM [6], Yi [65], Baichuan [61], Mistral [26], and DeepSeek [34]. These models demonstrate their strengths in multi-lingual comprehension, complex reasoning, and code generation tasks. By introducing these diverse models, our benchmark provides a more comprehensive picture of current LLMs' capabilities in hierarchical reasoning.

Hyperparameters. In the model setup, we make specific configurations for key parameters to ensure the model runs as intended. Specifically, we set the temperature to 0.0 to ensure the stability

and consistency of the generated content. We also fix the seed of the local model to 0 to facilitate the reproduction and validation of the results. In addition, we set the new token length to 2048, which allows the model to generate longer textual content to support complex tasks such as writing lengthy articles, generating detailed reports, or engaging in in-depth reasoning.

Evaluation. In experiments, we first generate a complete query by extracting a hierarchical structure from the benchmark dataset and filling it into a specific query template. Then, we prompt an LLM with the query to obtain the model’s response, which is subsequently formatted and compared with the correct answer. Accuracy is used as the primary metric to assess LLM performance, defined as $Accuracy = \frac{\#Correct}{\#Total}$.

5.2 LLMs’ Performance on HiBench

Overview. Table 2 presents the performance of the 20 most popular and powerful LLMs across tasks in our HiBench, categorized by model families and scenarios. Current LLMs achieve an average score of 40.48% on our HiBench, demonstrating the general level of hierarchical reasoning performance. Specifically, DeepSeek-V3 stands out with the highest performance, achieving a mean score of 65.56%. It surpasses the benchmark average by 62.0% and outperforms the most powerful closed-source LLM, GPT-4, by 10.2%. Although many LLMs demonstrate basic hierarchical reasoning capabilities in managing Multiple Tree, JSON, and Code scenarios, these LLMs still face challenges in more complex scenarios, especially in the practical aspects. These findings highlight the importance of continued model refinement to enhance LLMs’ behavior on more advanced and nuanced hierarchical tasks.

Scenario Performance. Figure 4 presents the performance of LLMs across various hierarchical reasoning tasks, categorized by different scenarios such as Multiple Tree, JSON, Code, Formula, Paper, and Binary Tree. The results show that LLMs perform well on tasks such as *Root Identification*, *Level Counting*, and *Node Attribute Recognition*, with performance values reaching up to 70%. These tasks, which require minimal reasoning complexity and exhibit clear structural patterns, allow LLMs to achieve high accuracy. In contrast, tasks like *Path Finding*, *Mirror Tree*, and *Calculation* are generally below 30% and pose significant challenges due to their requirements for multi-step reasoning, structural transformations, or numerical computations. Overall, the average performance across all tasks is 41.2%.

Capability Dimensions. As illustrated in Figure 1 (a), LLMs perform well in local relationship awareness, global structural understanding, and analytical reasoning but struggle with structural modifications and textual reasoning. DeepSeek-V3 leads overall, scoring 75.96% in local relationship awareness, 66.77% in global structural understanding, and 64.95% in analytical reasoning, yet still lags in structural modifications and textual reasoning. Similarly, GPT-4 and Qwen2.5-72B maintain strong local relationship awareness and analytical reasoning scores but steeply decline in structural modifications. Smaller models, such as Yi-1.5-9B and Phi-3.5-mini-3.8B, exhibit poor performance in structural manipulation and textual reasoning, while Baichuan-7B fails across all evaluation dimensions. The performance disparity highlights that, although

LLMs are proficient at recognizing and analyzing hierarchical structures, they struggle with structural modification and reasoning over hierarchical text, indicating significant room for improvement in these areas.

Structure Complexity Impact. Figure 5 presents the performance of LLMs across varying structure complexity levels. The results demonstrate that LLMs perform significantly better on simpler structures, with accuracy peaking for the *easy* structures in all scenarios. In contrast, as structure complexity increases, performance declines noticeably, with the *hard* structures yielding only 19.6% accuracy in the Binary Tree scenario and 37.0% in the JSON scenario. These findings suggest that while LLMs can effectively handle tasks with simpler structures and clear patterns, they struggle with more complex hierarchical structures, highlighting the need for further model development to address more challenging hierarchical structures.

5.3 Insightful Findings

5.3.1 Structure Representation Affect LLM Reasoning. As the mode of structural representation can affect LLMs’ ability to capture hierarchical information, we explore the impact of different representation formats on hierarchical reasoning performance. Specifically, we compare two representation modes: *edge* representation and *text tree* representation, over binary and multiple tree tasks. The edge representation encodes the hierarchical structure through a list of directed edges, while the text tree representation uses structure symbols to visually reflect the hierarchical organization. As shown in Figure 6 (a), LLMs utilizing text tree representation outperform those using edge representations on both binary tree and multi-tree tasks, achieving performance scores of 39.6% and 38.4%, respectively. These results suggest that input representation plays a significant role in shaping the hierarchical reasoning capabilities of LLMs.

The text tree representation conveys global hierarchical information. In contrast, the edge representation emphasizes only local connections, which may hinder its effectiveness in capturing the overall organizational pattern. It suggests that the text tree representation is better suited for tasks requiring a comprehensive understanding of the hierarchy.

5.3.2 Expression of Explicit Structures Improves LLM Comprehension. In practical aspects, such as Code and Paper scenarios, hierarchical structures may present with either implicit or explicit structural constraints. In the Code scenario, C++ utilizes explicit structural constraints where double braces mark each block, while Python relies on indentation to represent hierarchical relationships implicitly. In the Paper scenario, papers formatted in an XML-like style [8] offer clear structural indicators, whereas those based on plain text lack explicit hierarchical cues. Figure 6 (b) and (c) investigate how these kinds of structural constraints impact LLMs’ hierarchical reasoning capabilities. The results demonstrate that inputs with explicit structural constraints consistently outperform those with implicit or no structure. Specifically, C++ outperforms Python by 4.7% in the Code scenario, while structured inputs surpass unstructured ones by 1.1% in the Paper scenario. The superior performance could be attributed to the clarity and consistency of

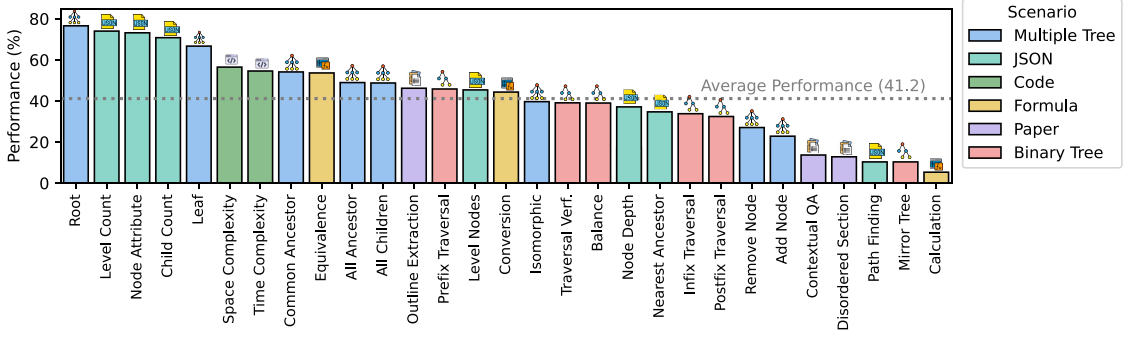


Figure 4: Average Performance of LLM Tasks across Capability Dimensions and Scenarios.

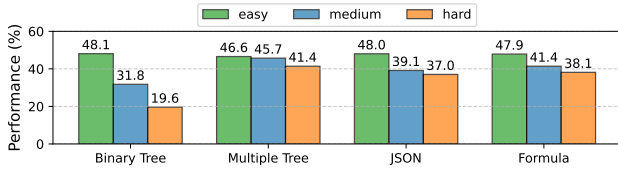


Figure 5: Impact of Structural Complexity on LLM Hierarchical Reasoning Capabilities.

explicit structural formats, such as C++’s strict syntactic rules and XML’s hierarchical markers, which define clear boundaries and reduce ambiguity, allowing LLMs to capture the hierarchical information more effectively. These findings highlight the critical role of explicit hierarchical structural constraints in enhancing the hierarchical reasoning behaviors of LLMs in practical aspects.

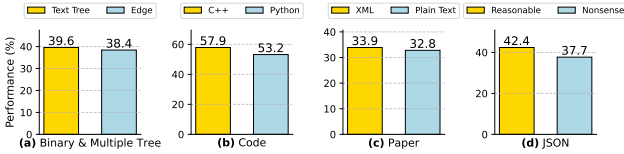


Figure 6: Impact of Input Mode on LLM Hierarchical Reasoning Capabilities.

5.3.3 Semantics Enhances LLM Hierarchical Structure Reasoning. To investigate whether LLMs have captured essential hierarchical reasoning capability from real-world corpora during pre-training, we compare their performance on real-world hierarchical structures and nonsense structures in the JSON scenario. The original JSON scenario is based on real-world hierarchical structures containing semantically meaningful values. We construct nonsense structures involving hierarchies with randomly shuffled labels or meaningless values for comparison. As shown in Figure 6 (d), LLMs achieve a significantly higher accuracy of 42.4% with real-world semantic structures compared to 37.7% with nonsense structures. This improvement suggests that LLMs have likely internalized basic

hierarchical structures from real-world data, enabling them to more effectively parse and reason over nested JSON hierarchies.

5.3.4 Impact of Chain-of-Thought. CoT improves performance in various tasks by prompting models to break down problems into step-by-step logical sequences. To evaluate its impact on LLM hierarchical reasoning, we conduct experiments of Qwen2.5-7B and QwQ-32B on Binary Tree and Multiple Tree scenarios. As shown in Table 3, CoT slightly improves Qwen2.5-7B’s accuracy in the Multiple Tree task by 3.33% but has minimal degradation on the Binary Tree task. However, for QwQ-32B, CoT leads to a 0.9% accuracy decrease in Binary Tree and a substantial 31.26% drop in Multiple Tree performance. These results indicate that CoT does not consistently enhance hierarchical reasoning and may introduce unnecessary steps that complicate intermediate reasoning.

Table 3: Performance Comparison of Models Using CoT Reasoning Versus Standard One.

	Binary Tree	Multiple Tree
Qwen2.5-7B w/ CoT	33.11	54.86
Qwen2.5-7B w/o CoT	33.06	58.19
Δ_{CoT}	-0.05 ↓	3.33 ↑
QwQ-32B w/ CoT	42.91	74.12
QwQ-32B w/o CoT	42.01	42.86
Δ_{CoT}	-0.9 ↓	-31.26 ↓

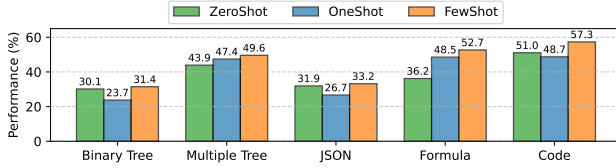
5.4 Potential Improvement

Given the nascent state of LLMs’ behaviors in hierarchical reasoning, we conduct a series of experiments, including ICL and instruction finetuning, to assess the effectiveness of these methods in improving their hierarchical reasoning performance.

Figure 7 demonstrates the impact of ICL on LLMs over our Hi-Bench, where zero-shot learning results in the lowest performance, one-shot learning shows moderate improvement, and few-shot learning significantly enhances LLMs’ hierarchical reasoning capabilities. This trend demonstrates that LLMs benefit from additional context provided by ICL, with more examples generally leading to substantially better performance across hierarchical reasoning

Table 4: Instruction Finetuning Performance on HiBench.

Aspects	Qwen2.5-7B		Llama-3.1-8B	
	Vanilla	Finetuned	Vanilla	Finetuned
Fundamental	42.32	65.74	25.89	63.59
Practical	54.17	61.05	37.20	55.60
Average	48.25	63.39	31.55	59.59

**Figure 7: Impact of ICL on Hierarchical Reasoning.**

tasks. However, in certain scenarios, such as Binary Tree, JSON, and Code, one-shot learning performs poorly compared to zero-shot learning. LLMs may develop an unintended inductive bias when only a single example is provided, leading to incorrect generalization. This effect is particularly pronounced when the example contains high-frequency patterns or structural repetitions, causing the model to overfit spurious correlations rather than accurately grasping the hierarchical relationships. It underscores the importance of instance diversity in few-shot settings to mitigate biases and enhance LLMs' hierarchical reasoning capabilities.

To further enhance the hierarchical reasoning performance of LLMs, we conduct instruction finetuning on Llama-3.1-8B and Qwen2.5-7B models using a well-designed instruction dataset with 14,623 examples targeted to hierarchical reasoning. The dataset emphasizes scenarios involving complex hierarchical structures, implicit structural representations, and counterfactual hierarchical configurations, aiming to strengthen LLMs' reasoning in areas where they currently underperform. As shown in Table 4, finetuning leads to significant performance gains across both fundamental and practical aspects. Specifically, Llama-3.1-8B improves from 31.55% to 59.59% on average, and Qwen2.5-7B rises from 48.25% to 63.39%. These results highlight the importance of finetuning LLMs to enhance hierarchical reasoning, particularly for complex tasks.

6 Conclusion

In this paper, we propose HiBench, the first benchmark dedicated explicitly to evaluating the hierarchical reasoning capabilities of LLMs. HiBench spans two key aspects, six scenarios, and 30 tasks, comprising 39,159 queries. Experimental results demonstrate that while existing LLMs show proficiency in basic hierarchical reasoning tasks, they still struggle with more complex hierarchical challenges. In addition to improving LLM performance through ICL, we demonstrate that finetuning small-scale LLMs on our constructed high-quality instruction dataset leads to significant improvements of up to 6.53% over the leading closed-source LLM GPT-4. These

findings highlight the potential for further advancements in enhancing LLMs' hierarchical reasoning capabilities.

Acknowledgments

The research described in this paper has been partly supported by General Research Funds from the Hong Kong Research Grants Council (project no. PolyU 15207322, 15200023, 15206024, and 152245 24), internal research funds from The Hong Kong Polytechnic University (project no. P0042693, P0048625, P0051361, P0052406, and P0052986).

References

- [1] Marah Abidin, Jyoti Aneja, Hany Awadallah, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219* (2024).
- [2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
- [3] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Conference on Empirical Methods in Natural Language Processing*. 3615–3620.
- [4] Anissa M Bettayeb, Manar Abu Talib, Al Zahraa Sobhe Altayasinah, and Fatima Dakalbab. 2024. Exploring the impact of ChatGPT: conversational AI in education. In *Frontiers in Education*, Vol. 9. Frontiers Media SA, 1379796.
- [5] Matthew M Botvinick. 2008. Hierarchical models of behavior and prefrontal function. *Trends in cognitive sciences* 12, 5 (2008), 201–208.
- [6] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297* (2024).
- [7] Avyay Casheekar, Archiit Lahiri, Kanishk Rath, Kaushik Sanjay Prabhakar, and Kathiravan Srinivasan. 2024. A contemporary review on chatbots, AI-powered virtual conversational agents, ChatGPT: Applications, open challenges and future research directions. *Computer Science Review* 52 (2024), 100632.
- [8] Tian-Yi Che, Xian-Ling Mao, Tian Lan, and Heyan Huang. 2024. A Hierarchical Context Augmentation Method to Improve Retrieval-Augmented LLMs on Scientific Papers. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 243–254.
- [9] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. 2008. Hierarchical structure and the prediction of missing links in networks. *Nature* 453, 7191 (2008), 98–101.
- [10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168* (2021).
- [11] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. *arXiv preprint arXiv:2004.07180* (2020).
- [12] Xinnan Dai, Haohao Qu, Yifei Shen, Bohang Zhang, Qihao Wen, Wenqi Fan, Dongsheng Li, Jiliang Tang, and Caihua Shan. 2025. How Do Large Language Models Understand Graph Patterns? A Benchmark for Graph Pattern Comprehension. In *The Thirteenth International Conference on Learning Representations*.
- [13] Xinnan Dai, Qihao Wen, Yifei Shen, Hongzhi Wen, Dongsheng Li, Jiliang Tang, and Caihua Shan. 2024. Revisiting the Graph Reasoning Ability of Large Language Models: Case Studies in Translation, Connectivity and Shortest Path. *arXiv preprint arXiv:2408.09529* (2024).
- [14] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. 2021. A Dataset of Information-Seeking Questions and Answers Anchored in Research Papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4599–4610.
- [15] Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From Explicit CoT to Implicit CoT: Learning to Internalize CoT Step by Step. *arXiv:2405.14838 [cs.CL]* <https://arxiv.org/abs/2405.14838>
- [16] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. 2024. A Survey on In-context Learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 1107–1128.
- [17] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. *arXiv:2103.10360 [cs.CL]* <https://arxiv.org/abs/2103.10360>

- [18] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).
- [19] Tao Feng, Chuanyang Jin, Jingyu Liu, Kunlun Zhu, Haoqin Tu, Zirui Cheng, Guanyu Lin, and Jiaxuan You. 2024. How Far Are We From AGI. *arXiv preprint arXiv:2405.10313* (2024).
- [20] Kurt W Fischer. 1980. A theory of cognitive development: The control and construction of hierarchies of skills. *Psychological review* 87, 6 (1980), 477.
- [21] Tim Genewein, Felix Leibfried, Jordi Grau-Moya, and Daniel Alexander Braun. 2015. Bounded rationality, abstraction, and hierarchical decision-making: An information-theoretic optimality principle. *Frontiers in Robotics and AI* 2 (2015), 27.
- [22] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793* (2024).
- [23] Jia Gong, Lin Geng Foo, Yixuan He, Hossein Rahmani, and Jun Liu. 2024. LLMs are good sign language translators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18362–18372.
- [24] Sagar Goyal, Eti Rastogi, Sree Prasanna Rajagopal, Dong Yuan, Fen Zhao, Jai Chintagunta, Gautam Naik, and Jeff Ward. 2024. Healai: A healthcare LLM for effective medical documentation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 1167–1168.
- [25] Tao He, Lizi Liao, Yixin Cao, Yuanxing Liu, Ming Liu, Zerui Chen, and Bing Qin. 2024. Planning Like Human: A Dual-process Framework for Dialogue Planning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 4768–4791.
- [26] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, et al. 2023. Mistral 7B. *arXiv:2310.06825* [cs.CL] <https://arxiv.org/abs/2310.06825>
- [27] Hanlei Jin, Yang Zhang, Dan Meng, Jun Wang, and Jinghua Tan. 2024. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *arXiv preprint arXiv:2403.02901* (2024).
- [28] Tomoyuki Kagaya, Thong Jing Yuan, Yuxuan Lou, Jayashree Karlekar, Sugiri Pranata, Akira Kinose, Koki Oguri, Felix Wick, and Yang You. 2024. RAP: Retrieval-Augmented Planning with Contextual Memory for Multimodal LLM Agents. In *NeurIPS 2024 Workshop on Open-World Agents*.
- [29] Roman Koshkin, Katsuhito Sudoh, and Satoshi Nakamura. 2024. TransLLaMa: LLM-based Simultaneous Translation System. In *Findings of the Association for Computational Linguistics: EMNLP* 2024. 461–476.
- [30] Minae Kwon, Hengyuan Hu, Vivek Myers, Siddharth Karamcheti, Anca Dragan, and Dorsa Sadigh. 2024. Toward grounded commonsense reasoning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5463–5470.
- [31] Jinyang Li, Binyuan Hui, Reynold Cheng, Bowen Qin, Chenhao Ma, Nan Huo, Fei Huang, Wenyu Du, Luo Si, and Yongbin Li. 2023. Graphix-t5: Mixing pre-trained transformers with graph-aware layers for text-to-sql parsing. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 13076–13084.
- [32] Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2023. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. *arXiv preprint arXiv:2305.13269* (2023).
- [33] Feng Lin, Dong Jae Kim, et al. 2024. When llm-based code generation meets the software development process. *arXiv preprint arXiv:2403.15852* (2024).
- [34] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Cheng-gang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).
- [35] Fang Liu, Yang Liu, Lin Shi, Houkun Huang, Ruifeng Wang, Zhen Yang, Li Zhang, Zhongqi Li, and Yuchi Ma. 2024. Exploring and evaluating hallucinations in llm-powered code generation. *arXiv preprint arXiv:2404.00971* (2024).
- [36] Yinquan Lu, Wenhao Zhu, Lei Li, Yu Qiao, and Fei Yuan. 2024. LLaMAX: Scaling Linguistic Horizons of LLM by Enhancing Translation Capabilities Beyond 100 Languages. In *Findings of the Association for Computational Linguistics: EMNLP* 2024. 10748–10772.
- [37] Linhao Luo, Jiaxin Ju, Bo Xiong, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Chatrule: Mining logical rules with large language models for knowledge graph reasoning. *arXiv preprint arXiv:2309.01538* (2023).
- [38] Zihan Luo, Xiran Song, Hong Huang, Jianxun Lian, Chenhao Zhang, Jinqi Jiang, Xing Xie, and Hai Jin. 2024. GraphInstruct: Empowering Large Language Models with Graph Understanding and Reasoning Capability. *arXiv preprint arXiv:2403.04483* (2024).
- [39] Sanwal Manish. 2024. An autonomous multi-agent llm framework for agile software development. *International Journal of Trend in Scientific Research and Development* 8, 5 (2024), 892–898.
- [40] Siddhant Meshram, Namit Naik, VR Megha, Tanmay More, and Shubhangi Kharche. 2021. Conversational AI: chatbots. In *2021 International Conference on Intelligent Technologies (CONIT)*. IEEE, 1–6.
- [41] Jürgen Mihm, Christoph H Loch, Dennis Wilkinson, and Bernardo A Huberman. 2010. Hierarchical structure and search in complex organizations. *Management science* 56, 5 (2010), 831–848.
- [42] Tala Mirzaei, Leila Amini, and Pouyan Esmaeilzadeh. 2024. Clinician voices on ethics of LLM integration in healthcare: A thematic analysis of ethical concerns and implications. *BMC Medical Informatics and Decision Making* 24, 1 (2024), 250.
- [43] Mortimer Mishkin, Wendy A Suzuki, David G Gadian, and Faraneh Vargha-Khadem. 1997. Hierarchical organization of cognitive memory. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 352, 1360 (1997), 1461–1467.
- [44] Sam Musker, Alex Duchnowski, Raphaël Millièvre, and Ellie Pavlick. 2024. Semantic Structure-Mapping in LLM and Human Analogical Reasoning. *arXiv preprint arXiv:2406.13803* (2024).
- [45] Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. 2024. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. 1–13.
- [46] OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. <https://openai.com/blog/chatgpt>
- [47] Benedikt Perak, Slobodan Beliga, and Ana Meštrović. 2024. Incorporating Dialect Understanding Into LLM Using RAG and Prompt Engineering Techniques for Causal Commonsense Reasoning. In *Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2024)*. 220–229.
- [48] Bharat Prakash, Tim Oates, and Tinoosh Moheeni. 2023. LLM Augmented Hierarchical Agents. *arXiv* (11 2023). doi:10.48550/arXiv.2311.05596 arXiv:2311.05596
- [49] Braden A Purcell and Roozbeh Kiani. 2016. Hierarchical decision processes that operate over distinct timescales underlie choice and changes in strategy. *Proceedings of the national academy of sciences* 113, 31 (2016), E4531–E4540.
- [50] Jianing Qiu, Kyle Lam, Guohao Li, Amish Acharya, Tien Yin Wong, Ara Darzi, Wu Yuan, and Eric J Topol. 2024. LLM-based agentic systems in medicine and healthcare. *Nature Machine Intelligence* 6, 12 (2024), 1418–1420.
- [51] Darryl W Schneider and Gordon D Logan. 2006. Hierarchical control of cognitive processes: switching tasks in sequences. *Journal of Experimental Psychology: General* 135, 4 (2006), 623.
- [52] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 645–654.
- [53] Qwen Team. 2024. QwQ: Reflect Deeply on the Boundaries of the Unknown. <https://qwenlm.github.io/blog/qwq-32b-preview/>
- [54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv:2302.13971* [cs.CL] <https://arxiv.org/abs/2302.13971>
- [55] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems* 36 (2023), 30840–30861.
- [56] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2024. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems* 36 (2024).
- [57] Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. 2024. Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding. In *The Twelfth International Conference on Learning Representations*.
- [58] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
- [59] Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. 2024. Agentless: Demystifying llm-based software engineering agents. *arXiv preprint arXiv:2407.01489* (2024).
- [60] Frank Xing. 2025. Designing Heterogeneous LLM Agents for Financial Sentiment Analysis. *ACM Trans. Manage. Inf. Syst.* 16, 1 (2025), 24.
- [61] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305* (2023).
- [62] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yeqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report. *arXiv:2407.10671* [cs.CL]

<https://arxiv.org/abs/2407.10671>

- [63] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115* (2024).
- [64] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems* 36 (2023), 11809–11822.
- [65] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652* (2024).
- [66] Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, et al. 2025. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems* 37 (2025), 137010–137045.
- [67] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. *Advances in neural information processing systems* 32 (2019).
- [68] Haopeng Zhang, Philip S Yu, and Jiawei Zhang. 2024. A Systematic Survey of Text Summarization: From Statistical Methods to Large Language Models. *arXiv preprint arXiv:2406.11289* (2024).
- [69] Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. 2020. GraphBert: Only Attention is Needed for Learning Graph Representations. *arXiv:2001.05140* [cs.LG] <https://arxiv.org/abs/2001.05140>
- [70] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic Chain of Thought Prompting in Large Language Models. In *The Eleventh International Conference on Learning Representations*.
- [71] Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, Xiaowei Yu, et al. 2024. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint arXiv:2409.18486* (2024).
- [72] Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Ed Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. 2025. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems* 37 (2025), 126032–126058.

A Dataset Statistics

Binary Tree Scenario As shown in Table 5, HiBench categorizes the Binary Tree Scenario into three difficulty levels based on the number of tree nodes and the depth of the tree. The figures listed for each level indicate the dataset size, the number of nodes, and the number of tree layers.

Table 5: Structure Statistics of Binary Tree Scenario.

Complexity	#Structure	#Node	#Layer	#Degree
Easy	48	2~15	2~4	2
Medium	54	16~255	5~7	2
Hard	36	256~511	8~9	2

Multiple Tree Scenario In the Multiple Tree scenario, HiBench designs six difficulty levels of varying sizes, differing in the number of nodes, node out-degree, and tree depth. As shown in Table 6, each difficulty level is characterized by four values: the dataset size, the total number of tree nodes, the number of tree layers, and the average out-degree of the nodes, respectively.

JSON Scenario The JSON dataset comprises seven types of questions derived from two categories of datasets: normal and nonsense. The normal part contains JSON structures with realistic and meaningful semantics, while the nonsense part uses randomly generated content. Details about the JSON dataset are illustrated in Table 7.

Table 6: Structure Statistics of Multiple Tree Scenario.

Complexity	#Structure	#Node	#Layer	#Degree
Easy	33	2~13	2~3	2~3
Medium-1	36	2~13	2~3	3~4
Medium-2	36	3~34	3~4	2~3
Hard-1	36	4~32	2~3	5~6
Hard-2	36	13~212	5~6	2~3

Table 7: Statistics of JSON Datasets.

Complexity	#Depth	#Width
Small	4	2
Medium-1	5	2
Medium-2	4	4
Large-1	6	2
Large-2	4	6

Code Scenario The dataset consists of 100 Python and 100 C++ scripts, each containing two questions about time and space complexity. Table 8 demonstrates the details of the Code dataset.

Table 8: Statistics of Code Understanding Datasets.

Programming Language	Python	C++
Script Content	LeetCode	LeetCode
Script Length	11 - 86 lines	6 - 37 lines
No. of scripts	100	100
Question no. of Type 1	100	100
Question no. of Type 2	100	100

Formula Scenario As shown in Table 9, we constructed a system of formulas of different complexities in the Formula task. These expressions are systematically assigned to Conversion, Calculation, and Equivalence. Specifically, the Conversion task contains 5,022 queries, the Calculation task contains 2,511 queries, and the Equivalence task contains 7,533 queries.

Table 9: Multidimensional Formula Complexity System.

Complexity	Easy	Medium	Hard
Type	int	-	float
Range	1 - 10	-50 - 50	-100 - 100
Symbolic	+ - * /	+ - * / ()	+ - * / () ^
Length	3	6	9

Paper Scenario This dataset focuses on academic paper understanding, encompassing three sub-tasks: *Outline Extraction*, *Disordered Section Identification*, and *Contextual Question Answering*. It aims to evaluate the capability of models to comprehend the structure and content of scholarly articles.