DS-MHP: Improving Chain-of-Thought through Dynamic Subgraph-Guided Multi-Hop Path

Yongqiang Liu[1,2], Qiyao Peng[1], Binrong Liu[1], Hongtao Liu[3], Xuewei Li[1], Wenjun Wang[1,2*]
[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Yazhou Bay Innovation Institute, Hainan Tropical Ocean University, Sanya Hainan, China
[3]Du Xiaoman Financial, Beijing, China
{lyq236, qypeng, binrong, htliu, lixuewei, wjwang}@tju.edu.cn

Abstract

Large language models (LLMs) excel in natural language tasks, with Chain-of-Thought (CoT) prompting enhancing reasoning through step-by-step decomposition. However, CoT struggles in knowledge-intensive tasks with multiple entities and implicit multi-hop relations, failing to connect entities systematically in zero-shot settings. Existing knowledge graph methods, limited by static structures, lack adaptability in complex scenarios. We propose DS-MHP, a zero-shot framework to enhance LLM reasoning in multi-entity relation tasks. DS-MHP operates in three stages: 1) constructing query-specific subgraphs by extracting entities and relations; 2) generating and refining multi-hop paths using a hybrid strategy of Breadth-First Search, greedy expansion, and LLM supplementation; and 3) guiding LLMs with subgraphs and paths, aggregating answers via majority voting. Evaluated on 12 datasets spanning commonsense, logical, symbolic, and arithmetic reasoning, DS-MHP outperforms baselines and state-of-the-art methods in nearly all benchmarks. It achieves overall average accuracy increases of 3.9% on Mistral-7B and 3.6% on GPT-3.5 Turbo compared to SOTA, with significant gains in logical and symbolic reasoning. Additionally, DS-MHP reduces runtime and LLM calls compared to SOTA, enhancing computational efficiency. These improvements demonstrate DS-MHP's superior reasoning accuracy, explainability, and efficiency in complex multi-entity tasks.

1 Introduction

Large language models (LLMs) (Hoffmann et al., 2022; Chowdhery et al., 2023; Touvron et al., 2023; OpenAI, 2023; DeepSeek-AI, 2025) have demonstrated remarkable capabilities across a wide range of natural language processing (NLP) tasks, such as question answering (Robinson et al., 2022; Li et al., 2024b; Singhal et al., 2025), machine translation (Moslem et al., 2023; Xu et al., 2023; Zhu et al., 2024), and information extraction (Dagdelen et al., 2024; Li et al., 2024c). Leveraging extensive pre-trained knowledge, these models generate coherent and contextually relevant responses. Chain-of-Thought (CoT) (Wei et al., 2022) has significantly enhanced LLMs' reasoning abilities by guiding them to decompose complex problems into sequential reasoning steps, outperforming traditional zero-shot and few-shot approaches in tasks requiring logical, symbolic, and arithmetic reasoning.

However, CoT-based methods face challenges in knowledge-intensive tasks involving multiple entities and implicit multi-hop relations. For example, in questions like "Which historical figure influenced a modern leader's policies through an intermediary event?". CoT may produce intermediate steps but struggles to systematically connect entities (e.g., historical figure, event, modern leader) across multiple relational hops in zero-shot settings without examples. While Named Entity Recognition (NER) (Wang et al., 2023b; Ye et al., 2024; Lu et al., 2024) and relation extraction (Wadhwa et al., 2023; Zhang et al., 2023a; Zhao et al., 2024) can identify entities and explicit relations, LLMs often lack structured mechanisms to infer implicit multi-step dependencies. Knowledge graph (KG)-based approaches, such as Paths-over-Graph (PoG) (Tan et al., 2025), rely on pre-defined KGs and few-shot prompts to explore multi-hop paths but are limited by static knowledge structures, restricted path diversity, and challenges in adapting to ambiguous multi-entity scenarios without dynamic implicit relation inference.

In this paper, we introduce DS-MHP, a novel framework designed to enhance LLM reasoning through dynamic subgraph-guided multi-hop path in complex multi-entity scenarios. DS-MHP operates in three stages: (1) Dynamic Subgraph Construction, where entities are extracted using zero-

*Corresponding Author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 11216–11230
November 4-9, 2025 ©2025 Association for Computational Linguistics

shot NER, and explicit and implicit relations are identified and scored for confidence to form a query-specific directed subgraph; (2) Multi-Hop Path Generation, which selects key entities based on semantic and structural relevance, generates diverse multi-hop paths using a hybrid strategy of Breadth-First Search (BFS), greedy expansion, and LLM supplementation, and refines them through merging, deduplication, semantic and LLM scoring, and subpath filtering; and (3) Question Answering, where the subgraph and paths are integrated into structured prompts to guide LLM, with answers aggregated via majority voting for robustness.

We evaluate DS-MHP on 12 widely adopted datasets covering commonsense, logical, symbolic, and arithmetic reasoning, using Mistral-7B (Albert et al., 2023) and GPT-3.5 Turbo (OpenAI, 2023). DS-MHP outperforms baselines and state-of-the-art (SOTA) method in nearly all benchmarks, achieving overall average accuracy increases of 3.9% on Mistral-7B and 3.6% on GPT-3.5 Turbo compared to SOTA. This indicates that dynamically constructing query-specific subgraphs and generating multi-hop paths can significantly enhance the reasoning capabilities and answer accuracy of LLMs. Our main contributions can be summarized as follows:

- We propose DS-MHP, a zero-shot framework that addresses complex multi-entity reasoning by dynamically constructing query-specific subgraphs and generating multi-hop paths, achieving robust and accurate answers across diverse reasoning tasks.

- DS-MHP builds dynamic query-specific subgraph via zero-shot NER, relation extraction and assessment, generates diverse multi-hop paths through a hybrid strategy of BFS, greedy expansion, and LLM supplementation, and delivers answers using structured prompts with majority voting.

- Empirical results demonstrate that DS-MHP achieves superior performance across four reasoning scenarios, with average accuracy gains of 3.9% on Mistral-7B and 3.6% on GPT-3.5 Turbo, particularly in logical and symbolic reasoning tasks.

2 Related Work

2.1 Chain-of-Thought Prompting

CoT prompting enhances the reasoning capabilities of LLMs by encouraging step-by-step problem decomposition (Wei et al., 2022). This approach guides LLMs to break down complex tasks into intermediate steps, improving performance in arithmetic, commonsense, and symbolic reasoning tasks. Zero-shot CoT (Kojima et al., 2022) further demonstrated that simple prompts, such as "Let's think step by step", enable LLMs to perform logical reasoning without demonstrations, achieving competitive results. Subsequent advances have refined CoT's applicability and efficiency. For instance, Auto-CoT (Zhang et al., 2023b) automates CoT construction by analyzing questions, reducing manual prompt engineering. CoT-SC (Wang et al., 2023c) introduces self-consistency, sampling multiple reasoning paths and selecting the most frequent outcome via majority voting to enhance robustness. Complex-CoT (Fu et al., 2023) estimates reasoning steps based on problem complexity, while Wang et al. (2023a) separates tasks into planning and solving phases to generate structured CoT answers. RE2 (Xu et al., 2024) improves question comprehension through iterative rephrasing, and Nash CoT (Zhang et al., 2024) optimizes multi-path inference using game-theoretic principles. More recently, ERA-CoT (Liu et al., 2024) incorporates entity relation analysis for multi-entity scenarios, and DeCoT (Wu et al., 2024) addresses logical inconsistencies via causal interventions.

However, CoT-based methods face significant challenges in knowledge-intensive tasks involving multiple entities and implicit multi-hop relations. These approaches often generate verbose or incoherent reasoning steps, particularly in zero-shot settings, where the lack of structured knowledge leads to increased computational overhead and reduced accuracy. Additionally, existing CoT methods struggle to systematically capture and reason over complex, implicit relations among entities, limiting their effectiveness in scenarios requiring deep contextual understanding.

2.2 KG-based LLM Reasoning

KGs provide structured representations of factual knowledge, significantly enhancing LLM reasoning capabilities (Pan et al., 2024). Early approaches embedded KG knowledge into LLMs during pretraining or fine-tuning, enabling models to leverage relational triples for tasks like question answering (Peters et al., 2019; Zhang et al., 2021; Li et al., 2024a; Luo et al., 2024). However, these embedding-based methods often compromised interpretability and required retraining for new domains. To address these limitations, prompt-based methods emerged, transforming KG triples into textual prompts to facilitate reasoning in natural language (Pan et al., 2024; Wen et al., 2024). While effective, these approaches frequently overlooked the structural richness of KGs, such as multi-hop relational paths. More recent advancements enable LLMs to directly navigate KGs, starting from an initial entity and iteratively exploring relation edges (Jiang et al., 2023; Sun et al., 2024; Ma et al., 2024). For instance, Think-on-Graph (ToG) (Sun et al., 2024) implements a graph-based reasoning loop to explore paths dynamically, while Paths-over-Graph (PoG) (Tan et al., 2025) constructs question-specific subgraphs from pre-defined KGs, employs few-shot prompting to guide multi-hop path exploration, and prunes paths using a three-stage Beam Search to ensure relevance.

Nevertheless, existing KG-based LLM reasoning methods exhibit notable limitations. Embedding-based approaches lack interpretability and flexibility, relying on static, domain-specific training. Prompt-based methods often fail to capture the structural details of KGs, limiting their ability to reason over complex relational paths. Navigation-based methods, typically starting from a single entity, struggle to incorporate multiple topic entities, leading to incomplete path exploration. Methods like PoG, which depend on pre-existing KGs, lack the ability to construct query-specific knowledge dynamically, restricting their effectiveness in complex multi-hop reasoning tasks. These limitations highlight the potential of adapting KG-inspired structured reasoning approaches to text-based inference, motivating the development of methods that dynamically construct knowledge representations from raw text.

3 Methodology

Problem Formulation. Given an input query q and context x, the objective is to predict the answer y by constructing a dynamic, query-specific subgraph $G_s = (E, R)$ with entity set E and relation set R, and deriving its multi-hop paths $P(G_s)$.We address this by maximizing the conditional probability of the answer given the subgraph and paths:

$$y = \arg\max_{y_i} P(y_i \mid G_s, P(G_s), q, x), \quad (1)$$

where y_i represents possible answer candidates (e.g., multiple-choice options or free-form responses). This formulation leverages G_s and $P(G_s)$ to guide LLM toward accurate answers across diverse contexts.

As shown in Figure 1, we introduce DS-MHP, a novel framework designed to enhance the model's understanding and reasoning of multi-entity relations in various NLP tasks. DS-MHP comprises three progressive stages: dynamic subgraph construction, multi-hop path generation, and question answering. The framework dynamically constructs query-specific knowledge subgraphs from text and leverages multi-hop paths to jointly learn explicit and implicit entity relations, filtering relevant knowledge to improve reasoning accuracy and adaptability.

3.1 Dynamic Subgraph Construction

This phase constructs a query-specific subgraph G_s by extracting and refining entities and relations from the input text, ensuring a robust representation of context-specific knowledge.

NER. The framework employs the information extraction capabilities of LLM in a zero-shot setting to identify entities from the query q and context x. The LLM generates n_p reasoning paths, each producing a candidate entity list extracted from its output. Entities undergo normalization by removing parenthetical content and converting to lowercase for consistency. Candidate entities are aggregated by removing duplicates using a set-based approach, yielding the entity set E. This method mitigates entity ambiguity and eliminates redundancy, providing a clean and reliable entity foundation.

Relation Extraction. The framework extracts both explicit and implicit relations among entities in E within a zero-shot setting to form the relation set R.

(1) **Explicit Relations.** The LLM's contextual understanding is leveraged by prompting it with E, q, and x to generate n_p reasoning paths. Each path produces candidate relational triples (e_i, r, e_j), where $e_i, e_j \in E$ and r is a concise relation phrase (e.g., "is_a", "part_of", "locate_in"). These triples are aggregated by removing duplicates to form the explicit relation set R_{ext}, capturing direct connections explicitly stated in the text.

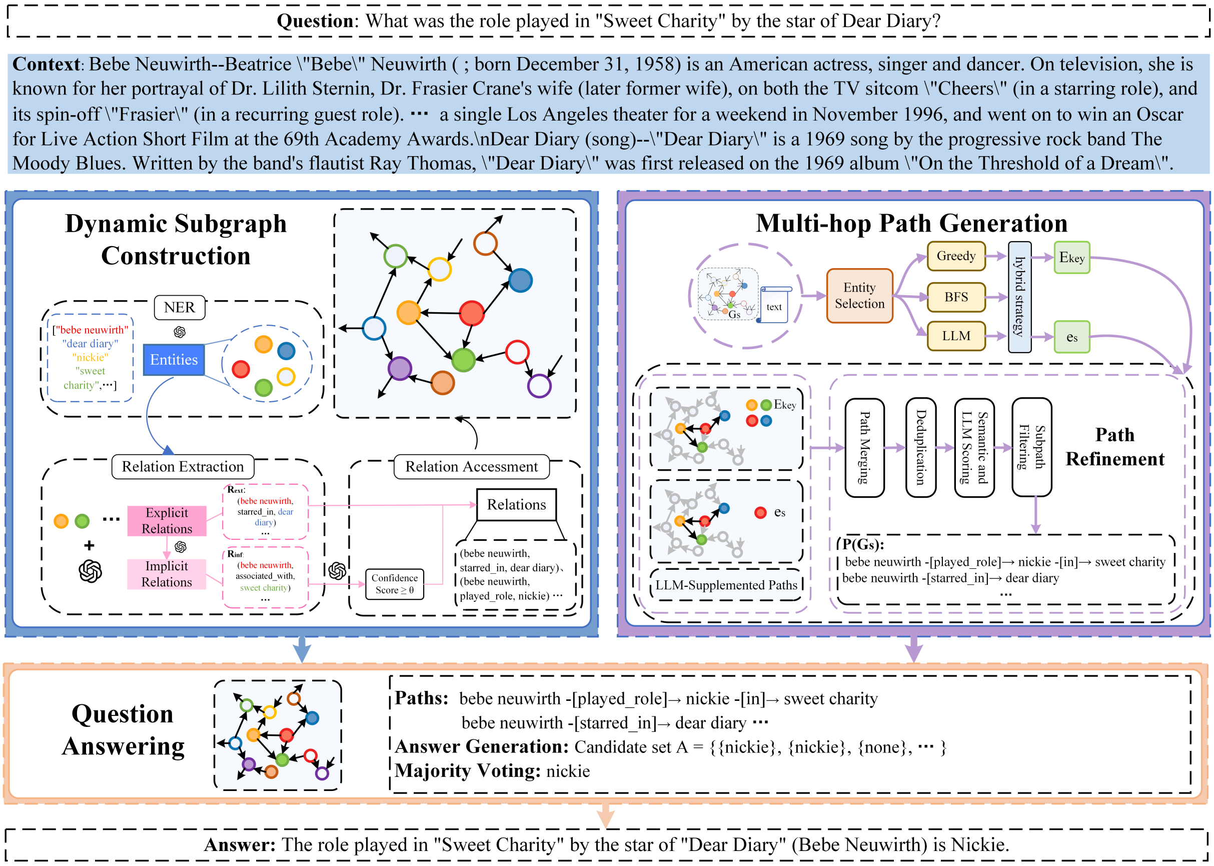

Figure 1: Overview of the DS-MHP framework, illustrating the three-stage process of dynamic subgraph construction, multi-hop path generation, and question answering.

(2) **Implicit Relations.** Implicit relations, not explicitly stated, are inferred based on E, R_{ext}, and the context x. The LLM generates n_p reasoning paths, producing up to n_{inf} candidate implicit triples per entity pair. These triples are aggregated by removing duplicates, retaining only syntactically valid triples (i.e., those with $e_i, e_j \in E$ and a plausible r) to form the implicit relation set:

$$R_{\text{inf}} = \{(e_i, r, e_j) \mid e_i, e_j \in E\}. \tag{2}$$

This step enriches the subgraph with both direct and inferred knowledge, deducible from context and explicit relations.

Relation Assessment. The LLM evaluates the confidence of implicit triples in R_{inf} as a scoring agent. Each triple (e_i, r, e_j) is assessed using a prompt incorporating E, R_{ext}, and x, producing a confidence score $s(e_i, r, e_j) \in [0, 1]$. Triples with a score exceeding a threshold θ_r are merged into R_{ext} to form the final relation set:

$$\begin{aligned} R = R_{\text{ext}} \cup \{(e_i, r, e_j) \mid (e_i, r, e_j) \in R_{\text{inf}}, \\ s(e_i, r, e_j) \geq \theta_r\}. \end{aligned} \tag{3}$$

The subgraph $G_s = (E, R)$ is constructed as a directed graph, filtering out unreliable inferences to ensure a high-quality knowledge representation tailored to the query.

3.2 Multi-Hop Path Generation

This phase generates a refined set of multi-hop paths $P(G_s)$ from the subgraph G_s to facilitate structured reasoning over complex relations, enabling LLM to systematically explore dependencies. The process integrates multiple path generation strategies with pruning techniques, to ensure diversity and relevance.

Entity Selection. The framework selects entities to anchor path generation (Algorithms 1 in Appendix A), balancing structural connectivity and semantic relevance:

(1) **Key Entities (E_{key}):** Entities appearing in the query q or context x (case-insensitive) are identified. For each entity $e \in E$, a combined score is computed:

$$s_{\text{key}}(e) = s_{\text{sem}}(e, q) + w_q \mathbf{I}_q(e) + w_x \mathbf{I}_x(e), \tag{4}$$

where $s_{\text{sem}}(e, q) = \frac{\mathbf{v}_e \cdot \mathbf{v}_q}{\|\mathbf{v}_e\| \|\mathbf{v}_q\|}$ is the cosine similarity (normalized to $[0, 1]$) between embeddings of e and q using an embedding model M (all-MiniLM-L6-v2[1]), $\mathbf{I}_q(e) = 1$ if $e \in q$, else 0, and $\mathbf{I}_x(e) = 1$ if $e \in x$, else 0, with weights $w_q = 1.0$, $w_x = 0.5$. The top k entities with the highest $s_{\text{key}}(e)$ form $E_{\text{key}} \subseteq E$, capturing query and context-specific focal points.

(2) **Starting Entity** (e_s): A single entity e_s is selected to explore broader graph connectivity, scored as:

$$s_{\text{start}}(e) = \frac{\frac{d_{\text{out}}(e)}{d_{\max}} + s_{\text{sem}}(e, q)}{2}, \tag{5}$$

where $d_{\text{out}}(e)$ is the out-degree of e in G_s, $d_{\max} = \max_{e' \in E} d_{\text{out}}(e')$ normalizes the out-degree, and $s_{\text{sem}}(e, q)$ is defined as above. The starting entity is $e_s = \arg\max_{e \in E} s_{\text{start}}(e)$.

Path Generation. Paths are generated in three complementary phases using a hybrid strategy of BFS, greedy expansion, and LLM supplementation (Algorithms 2 in Appendix A), constrained by a fixed hop limit $h_{\max} = 5$, to capture diverse reasoning trajectories:

(1) **Key Entity Joint Paths**: For each consecutive pair (e_i, e_{i+1}) in E_{key}, if a path exists in G_s, the shortest path from e_i to e_{i+1} (computed via BFS) is added to $P(G_s)$ if its length satisfies $l(p) \leq h_{\max}$, reflecting explicit relations in q and x.

(2) **Starting Entity Paths**: From e_s, shortest paths (via BFS) to all reachable nodes in G_s are computed, forming an initial path set P_{init} with $l(p) \leq h_{\max}$. Paths with a semantic similarity score $s_{\text{sem}}(p, q) > \theta_{\text{sem}}$ (via M) are greedily extended by appending neighboring nodes if the extended path's score exceeds a fraction α of the original score, exploring deeper dependencies within $h_{\max}$.

(3) **LLM-Supplemented Paths**: The LLM is prompted with e_s, E, R, q, and x to suggest up to three additional reasoning paths, validated against G_s (i.e., entities in E and consecutive entities connected via R) and constrained to $l(p) \leq h_{\max}$, enriching $P(G_s)$ with inferred connections.

Path Refinement. The framework ensures a concise and relevant $P(G_s)$ through the following steps:

(1) **Path Merging**: Paths with identical start and end entities are merged. For single-hop paths, relations are fused (e.g., r_1 and r_2 into r_1 AND r_2); for multi-hop paths, the longest path is retained to preserve richer semantics in G_s.

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

(2) **Deduplication**: Paths are deduplicated, retaining only unique sequences with no repeated entities.

(3) **Semantic and LLM Scoring**: Each path p is evaluated with a combined metric:

$$s(p) = s_{\text{sem}}(p, q) + s_{\text{LLM}}(p, q, x), \tag{6}$$

where $s_{\text{sem}}(p, q)$ is the cosine similarity (via M) between the path string (entities and relations) and q, and $s_{\text{LLM}}(p, q, x) \in [0, 1]$ is LLM's relevance score, computed by prompting LLM to assess the path's relevance to q and x.

(4) **Subpath Filtering**: Paths are sorted by $s(p)$ in descending order, retaining the top n_{path} paths via Beam Search. Subpaths subsumed by longer, higher-scoring paths are discarded, ensuring non-redundancy in $P(G_s)$.

This multi-phase approach ensures $P(G_s)$ contains non-redundant, semantically relevant multi-hop paths tailored to the query and context, enhancing LLM's reasoning over complex relations.

3.3 Question Answering

We generate answer candidates using G_s and $P(G_s)$ in a zero-shot setting with LLM. The subgraph and paths are formatted into a structured natural language prompt, including G_s, $P(G_s)$, q, and x, enhancing CoT reasoning by systematically exploring multi-hop relations. For each path $p \in P(G_s)$, combined with G_s, LLM produces a candidate answer, forming set A. A fixed number of candidate answers are generated, and the final answer is selected using a majority voting strategy (Wang et al., 2023c):

$$A = \{\pi(p, G_s, q, x) \mid p \in P(G_s)\}, \tag{7}$$

$$y = \arg\max_{y_i} \text{Count}(y_i, A), \tag{8}$$

where π is LLM's prediction function, and $\text{Count}(y_i, A)$ is the frequency of candidate y_i in A. If the top answer's count is below a support threshold τ or a tie occurs among top answers, additional answers are iteratively generated (up to n_{attempt} attempts) using the same prompt, updating A until a reliable y is determined. This approach ensures robust answer selection for the question.

4 Experimental Setup

4.1 Datasets and Models

We consider four reasoning scenarios, i.e., commonsense reasoning, symbolic reasoning, logical

Model	Method	Common			Logical			Symbolic			Arithmetic			Overall Avg.
		StrategyQA	CSQA	ARC-C	LogiQA	ReClor	AR-LSAT	Date	Obj-Track	Letter	AQuA	GSM8K	SVAMP	
Mistral-7B	Vanilla LLM	59.7	67.8	73.5	44.5	51.8	19.9	31.2	29.2	0.0	20.5	6.4	47.4	37.7
	CoT	60.5	71.6	74.7	47.4	54.2	23.6	33.6	32.8	0.0	22.1	8.2	50.5	39.9
	CoT-SC@5	61.2	73.1	76.2	50.3	56.9	25.8	34.8	34.4	0.2	24.4	10.1	53.6	41.8
	Auto-CoT	60.8	72.8	75.8	48.6	55.4	25.3	34.0	33.2	0.0	23.2	9.4	51.9	40.9
	Complex-CoT	59.5	71.5	73.7	50.1	56.5	25.5	34.4	33.6	0.0	24.0	13.5	55.1	41.5
	PS	60.7	72.4	74.9	48.1	54.7	24.1	34.0	33.2	0.2	24.4	12.3	54.2	41.1
	PS+	61.3	72.7	75.6	48.5	55.3	25.7	34.4	34.0	0.4	25.2	12.8	55.3	41.8
	RE2	62.1	73.2	76.5	49.9	56.4	26.3	35.2	34.8	0.6	24.0	13.0	54.5	42.2
	ERA-CoT	64.2	74.8	78.6	51.2	57.6	27.8	36.8	36.0	0.4	25.6	14.5	55.2	43.6
	DS-MHP	**67.5**	**77.3**	**82.7**	**54.8**	**61.8**	**32.7**	**42.4**	**41.2**	**1.2**	**30.7**	**17.8**	**59.8**	**47.5**
GPT-3.5 Turbo	Vanilla LLM	65.6	72.3	82.9	28.5	52.5	21.1	45.2	35.6	3.0	31.9	52.6	77.4	47.4
	CoT	63.4	77.4	80.7	36.5	56.8	17.4	47.6	33.6	3.2	59.8	70.5	79.8	52.2
	CoT-SC@5	65.3	78.5	84.5	38.3	60.7	22.3	48.8	36.4	3.8	**66.5**	74.8	83.2	55.3
	Auto-CoT	64.8	77.8	81.5	38.7	61.5	22.5	46.8	35.2	3.2	54.7	77.4	84.5	54.1
	Complex-CoT	64.4	76.4	80.8	38.8	61.8	22.4	47.2	35.6	3.6	57.4	**80.2**	**86.3**	54.6
	PS	65.9	77.7	81.2	37.5	58.6	21.6	46.4	34.4	3.2	53.5	76.6	83.3	53.3
	PS+	66.4	77.3	82.4	38.9	61.2	22.8	47.2	35.6	3.4	52.3	76.1	82.7	53.9
	RE2	67.3	79.5	83.2	39.2	62.7	23.5	46.8	36.0	3.2	53.8	76.7	83.5	54.6
	ERA-CoT	71.5	83.5	83.4	45.3	64.5	24.9	48.0	36.2	3.4	56.9	79.8	82.2	56.6
	DS-MHP	**74.8**	**86.7**	**88.3**	**50.8**	**68.9**	**28.8**	**51.6**	**40.4**	**4.2**	62.5	79.5	86.1	**60.2**

Table 1: Main experimental results. The best results are highlighted in bold. We use accuracy as the evaluation metric. CoT-SC@5 represents retrieving five CoT reasoning chains to make majority votes.

reasoning, and arithmetic reasoning. Specifically, for commonsense reasoning, we use CommonsenseQA (CSQA) (Talmor et al., 2019), StrategyQA (Geva et al., 2021) and ARC-Challenge (ARC-C) (Clark et al., 2018); for symbolic reasoning, we use Date Understanding (Date), Object Tracking (Obj_Track) (Suzgun et al., 2023) and Last Letters (Letter) (Wei et al., 2022); for logical reasoning, we use LogiQA (Liu et al., 2021), ReClor (Yu et al., 2020), and AR-LSAT (Wang et al., 2022); for arithmetic reasoning, we use AQuA (Ling et al., 2017), GSM8K (Cobbe et al., 2021) and SVAMP (Patel et al., 2021). The details of dataset statistics are in Appendix C. For models, we use Mistral-7B (Albert et al., 2023) and GPT-3.5 Turbo (175B) (OpenAI, 2023).

4.2 Baselines

To comprehensively evaluate our method, we compare DS-MPR with the leading CoT methods baselines: Vanilla LLM, CoT (Wei et al., 2022), CoT-SC (Wang et al., 2023c), Auto-CoT (Zhang et al., 2023b), Complex-CoT (Fu et al., 2023), PS and PS+ (Wang et al., 2023a), RE2 (Xu et al., 2024) and ERA-CoT (SOTA)(Liu et al., 2024). The simple introduction of baselines is in Appendix B.

4.3 Implementation

We access GPT-3.5 Turbo through the OpenAI (OpenAI, 2023) API (gpt-3.5-turbo-0301) and utilize Mistral-7B with its default model parameters from the original implementation (Albert et al., 2023). The details of parameter settings are in Appendix D. To ensure reliability, we conduct five rounds of experiments for each dataset, reporting average scores. For evaluation, we use Exact Match (EM) and Accuracy (Acc) metrics. Further details are provided in Appendix E. The experiments are conducted on a single NVIDIA A100-80G GPU for each method.

5 Experiments

5.1 Main Results

Table 1 presents the main experimental results. **DS-MHP achieves superior performance, outperforming all baselines, including the SOTA ERA-CoT, on Mistral-7B across all 12 datasets and on GPT-3.5 Turbo for 9 out of 12 datasets, with overall average accuracies of 47.5% and 60.2%, respectively.** This indicates that through dynamic subgraph construction and multi-hop path generation, LLMs could make better predictions and enhance their performance. DS-MHP demonstrates robust performance across diverse reasoning tasks, with particularly notable gains in logical and symbolic reasoning, while maintaining strong results in commonsense and arithmetic tasks. Our source code is public at https://github.com/casanovalauz/DS-MHP.

Commonsense Reasoning. DS-MHP achieves average accuracies of 75.8% on Mistral-7B and 83.3% on GPT-3.5 Turbo across StrategyQA, CSQA, and ARC-C, surpassing ERA-CoT by 4.4% and 4.8%, respectively. Significant improvements are observed on ARC-C and CSQA. Compared to CoT, DS-MHP improves by 7.1% on Mistral-7B.

DS-MHP's dynamic subgraph approach effectively filters irrelevant reasoning paths, enhancing robustness in commonsense reasoning tasks.

Logical Reasoning. On LogiQA, ReClor, and AR-LSAT, DS-MHP attains average accuracies of 49.8% (Mistral-7B) and 49.5% (GPT-3.5 Turbo), outperforming ERA-CoT by 9.5% and 10.2%, respectively. The largest gain is on AR-LSAT, where multi-entity reasoning navigates complex logical structures. Compared to CoT, DS-MHP achieves an 8.7% improvement on Mistral-7B. These results highlight DS-MHP's strength in capturing intricate relational dependencies through structured multi-hop paths.

Symbolic Reasoning. DS-MHP excels on Date, Obj-Track, and Letter, with average accuracies of 28.3% (Mistral-7B) and 32.1% (GPT-3.5 Turbo), surpassing ERA-CoT by 16.0% and 9.9%, respectively. Notable gains are seen on Date, which requires temporal reasoning, and Obj-Track, which involves tracking multi-entity interactions. On Letter, which demands complex sequence processing, DS-MHP achieves 1.2% (vs. 0.4%) on Mistral-7B and 4.2% (vs. 3.4%) on GPT-3.5 Turbo. Despite the task's difficulty, DS-MHP's improvements demonstrate its capability to handle intricate pattern recognition through dynamic subgraph-based reasoning.

Arithmetic reasoning. For AQuA, GSM8K, and SVAMP, DS-MHP achieves average accuracies of 36.1% (Mistral-7B) and 76.0% (GPT-3.5 Turbo), outperforming ERA-CoT by 13.5% and 4.1%, respectively. DS-MHP excels on AQuA, leveraging contextual entity analysis for complex problems. However, on GPT-3.5 Turbo, it slightly trails CoT-SC@5 on AQuA (62.5% vs. 66.5%) and Complex-CoT on GSM8K (79.5% vs. 80.2%) and SVAMP (86.1% vs. 86.3%). Compared to CoT, DS-MHP improves by 14.7% on Mistral-7B. These results demonstrate that DS-MHP's subgraph-based reasoning enhances performance on tasks with relational complexity, but numerical computation-heavy tasks benefit less compared to sampling-based methods.

5.2 Ablation Study

We evaluate the contributions of DS-MHP's core components by conducting an ablation study comparing the complete DS-MHP method with three variants: (1) Subgraph Only, which uses only the dynamic subgraph construction module without multi-hop path generation; (2) Multi-Hop Paths Only, which provides LLM with only the multi-hop paths $P(G_s)$ during question answering, without the dynamic subgraph G_s; and (3) No Majority Voting, which removes the majority voting mechanism and uses the answer from the most reliable path instead.

Table 2 summarizes the results. **The complete DS-MHP method consistently outperforms all ablated variants, confirming the importance of integrating dynamic subgraph construction, multi-hop path generation, and majority voting.** On Mistral-7B, DS-MHP achieves an overall average accuracy of 47.5%, compared to 44.6% for Subgraph Only (2.9% drop), 43.5% for Multi-Hop Paths Only (4.0% drop), and 46.0% for No Majority Voting (1.5% drop). On GPT-3.5 Turbo, DS-MHP attains 60.2%, surpassing Subgraph Only (57.7%, 2.5% drop), Multi-Hop Paths Only (56.7%, 3.5% drop), and No Majority Voting (59.0%, 1.2% drop). The smaller performance drops on GPT-3.5 Turbo reflect its greater robustness compared to Mistral-7B.

Subgraph Only. The Subgraph Only variant, which relies solely on the dynamic subgraph G_s without multi-hop path generation, shows reduced performance, particularly in logical and symbolic reasoning. This indicates that while G_s provides a structured knowledge foundation, the absence of multi-hop path exploration limits the ability to navigate complex relational dependencies.

Multi-Hop Paths Only. The Multi-Hop Paths Only variant, which provides only the multi-hop paths $P(G_s)$ to LLM during question answering without the dynamic subgraph G_s, exhibits the largest performance drop (4.0% on Mistral-7B; 3.5% on GPT-3.5 Turbo). The lack of G_s's comprehensive knowledge structure restricts the relational context available for reasoning, significantly impacting symbolic and logical reasoning tasks, where multi-entity interactions are critical.

No Majority Voting. The No Majority Voting variant, which uses the answer from the most reliable path instead of aggregating answers from multiple paths via majority voting, reduces overall accuracy by 1.5% on Mistral-7B and 1.2% on GPT-3.5 Turbo. Declines are notable in logical and symbolic reasoning, indicating that majority voting enhances answer reliability by leveraging multiple paths to mitigate errors.

These results highlight the synergistic effect of DS-MHP's components. Dynamic subgraph construction provides a robust knowledge founda-

Model	Variant	Commonsense	Logical	Symbolic	Arithmetic	Overall Avg.
Mistral-7B	Subgraph Only	73.2	46.8	25.5	33.2	44.6
	Multi-Hop Paths Only	72.8	46.0	24.5	32.8	43.5
	No Majority Voting	74.2	48.2	26.8	34.8	46.0
	Complete DS-MHP	**75.8**	**49.8**	**28.3**	**36.1**	**47.5**
GPT-3.5 Turbo	Subgraph Only	81.2	47.0	29.8	73.5	57.7
	Multi-Hop Paths Only	80.8	46.5	29.0	73.0	56.7
	No Majority Voting	82.0	48.0	31.0	75.0	59.0
	Complete DS-MHP	**83.3**	**49.5**	**32.1**	**76.0**	**60.2**

Table 2: Ablation study results. The best results are highlighted in bold. We use average accuracy to compute accuracies across datasets in each reasoning scenario.

tion, multi-hop path generation enables complex relational reasoning, and majority voting ensures reliable answer aggregation. The moderate performance drops in ablated variants, with larger declines on Mistral-7B than on GPT-3.5 Turbo, demonstrate DS-MHP's robustness and the complementary contributions of each component to its superior performance across diverse reasoning tasks.

5.3 Efficiency Comparison

We compare the computational efficiency of DS-MHP and ERA-CoT on AR-LSAT and Obj-Track datasets, measuring runtime (seconds) and LLM calls per question using Mistral-7B and GPT-3.5 Turbo. Table 3 summarizes the results. DS-MHP consistently requires less runtime and fewer LLM calls than ERA-CoT across both datasets and models, with larger savings on AR-LSAT's complex reasoning tasks due to efficient relation assessment and path refinement techniques that prune unreliable inferences early. These efficiency gains align with the improved accuracy reported in Table 1. Obj-Track's simpler structure results in lower overall costs compared to AR-LSAT's demanding logical reasoning. These results highlight DS-MHP's balance of efficiency and accuracy in multi-entity reasoning tasks.

Model	Method	AR-LSAT		Obj-Track	
		Time (s)	Calls	Time (s)	Calls
Mistral-7B	ERA-CoT	4.4	8.6	2.0	4.0
	DS-MHP	3.8	7.2	1.8	3.5
GPT-3.5 Turbo	ERA-CoT	4.9	9.4	2.4	4.4
	DS-MHP	4.2	7.8	2.1	3.8

Table 3: Efficiency comparison of DS-MHP, and ERA-CoT on AR-LSAT and Obj-Track datasets, reporting average runtime (seconds) and LLM calls per question for Mistral-7B and GPT-3.5 Turbo.

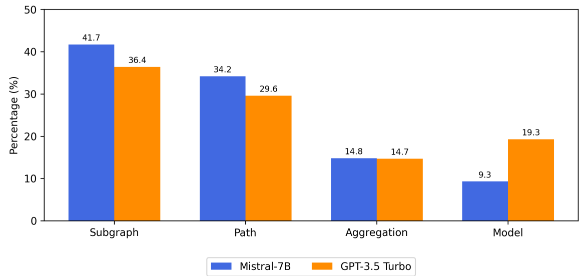

Figure 2: Error distribution for DS-MHP on AR-LSAT and Obj-Track, based on manual analysis of error cases. Percentages reflect the proportion of errors attributed to each category.

5.4 Error Analysis

We analyze DS-MHP's errors to identify its limitations and guide future improvements. Focusing on logical (AR-LSAT) and symbolic (Obj-Track) reasoning tasks, where DS-MHP shows significant gains in **Main Results** but notable declines in **Ablation Study**. Errors are manually inspected and categorized into subgraph construction, path generation, answer aggregation, and model reasoning issues.

Figure 2 shows that subgraph construction and path generation errors dominate, accounting for 75.9% of errors on Mistral-7B and 66.0% on GPT-3.5 Turbo. In AR-LSAT, DS-MHP often fails when the dynamic subgraph G_s omits implicit relations. For example, in a question requiring the inference "If A, then B; not B, therefore not A," Mistral-7B's subgraph misses the conditional relation, leading to an incorrect answer. GPT-3.5 Turbo mitigates some errors by inferring missing relations, but still struggles with severely incomplete subgraphs, aligning with the Multi-Hop Paths Only variant's 4.0% drop on Mistral-7B (vs. 3.5% on GPT-3.5 Turbo).

In Obj-Track, path generation errors are prevalent, where multi-hop paths $P(G_s)$ include irrelevant relations. For instance, in a question tracking object ownership after swaps (e.g., "Alice swaps a red ball with Bob, who swaps with Charlie"), DS-MHP generates a path connecting the ball to an irrelevant entity (e.g., "ball $\rightarrow$ table"), causing errors on both models. This corresponds to the Subgraph Only variant's 2.9% drop on Mistral-7B (vs. 2.5% on GPT-3.5 Turbo). Answer aggregation errors, less frequent, occur in tasks like LogiQA when majority voting favors a low-quality path, contributing to the No Majority Voting variant's 1.5% drop on Mistral-7B.

Subgraph construction and path generation errors are the primary limitations of DS-MHP in logical and symbolic reasoning tasks. Mistral-7B's higher error rates, particularly in subgraph construction, reflect its greater reliance on DS-MHP's components compared to GPT-3.5 Turbo.

6 Conclusion

In this paper, we propose DS-MHP to address the limitations of LLMs in complex knowledge reasoning and open-domain question answering tasks. By leveraging dynamic subgraphs, multi-hop paths, and majority voting, DS-MHP excels in diverse reasoning scenarios, with particularly notable gains in logical and symbolic reasoning tasks. Extensive experiments demonstrate its superior performance, along with reduced runtime and LLM calls for enhanced computational efficiency. These results validate the effectiveness of DS-MHP's modular design in improving both reasoning accuracy and practical applicability across various reasoning scenarios.

Limitations

We acknowledge that DS-MHP struggles with incomplete subgraph construction and imprecise path generation, often missing implicit relations or including irrelevant connections, which limits its effectiveness. Its sensitivity to noisy or incomplete input reduces robustness in diverse scenarios. Future improvements could incorporate external knowledge bases to enhance subgraphs, develop context-aware path scoring to refine paths, and improve input processing to boost robustness, thereby strengthening DS-MHP's performance across varied reasoning tasks.

Acknowledgments

This work was supported by the National Key R&D Plan of China (Grant No. 2022YFC2602305), the National Natural Science Foundation of China (No. 62472305), the Key R&D projet in Hainan province, China (No. ZDYF2024SHFZ051), and the Hainan Tropical Ocean Institute Yazhou Bay Innovation Research Institute Major Science and Technology Plan Project (No. 2023CXYZD001).

References

Q. Jiang Albert, Sablayrolles Alexandre, Mensch Arthur, Bamford Chris, Singh Chaplot Devendra, de las Casas Diego, Bressand Florian, Lengyel Gianna, Lample Guillaume, Saulnier Lucile, Renard Lavaud Lélio, Lachaux Marie-Anne, Stock Pierre, Le Scao Teven, Lavril Thibaut, Wang Thomas, Lacroix Timothée, and El Sayed William. 2023. Mistral 7b. *arXiv preprint arXiv.2310.06825*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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

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*.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024a. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *The Twelfth International Conference on Learning Representations*.

Zhenyu Li, Sunqi Fan, Yu Gu, Xiuxing Li, Zhichao Duan, Bowen Dong, Ning Liu, and Jianyong Wang. 2024b. Flexkbqa: A flexible llm-powered framework for few-shot knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18608–18616.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Lixiang Lixiang, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024c. Knowcoder: Coding structured knowledge into llms for universal information extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8758–8779.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2021. Logiqa: a challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3622–3628.

Yanming Liu, Xinyue Peng, Tianyu Du, Jianwei Yin, Weihao Liu, and Xuhong Zhang. 2024. Era-cot: Improving chain-of-thought through entity relationship analysis. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8780–8794.

Jinghui Lu, Yanjie Wang, Ziwei Yang, Xuejing Liu, Brian Mac Namee, and Can Huang. 2024. Padellmner: parallel decoding in large language models for named entity recognition. *Advances in Neural Information Processing Systems*, 37:117853–117880.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*.

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, and Jian Guo. 2024. Think-on-graph 2.0: Deep and interpretable large language model reasoning with knowledge graph-guided retrieval. *arXiv preprint arXiv:2407.10805*. Version 7.

Yasmin Moslem, Rejwanul Haque, John D Kelleher, and Andy Way. 2023. Adaptive machine translation with large language models. *arXiv preprint arXiv:2301.13294*. Version 3.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*. Version 6.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2022. Leveraging large language models for multiple choice question answering. *arXiv preprint arXiv:2210.12353*. Version 3.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, Darlene Neal, Qazi Mamunur Rashid, Mike Schaekermann, Amy Wang, Dev Dash, Jonathan H Chen, Nigam H Shah, Sami Lachgar, Philip Andrew Mansfield, and

16 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, pages 1–8.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *ACL (Findings)*, pages 13003–13051.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, and Wenjie Zhang. 2025. Paths-over-graph: Knowledge graph enpowered large language model reasoning. In *THE WEB CONFERENCE 2025*.

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 preprint arXiv:2302.13971*.

Somin Wadhwa, Silvio Amir, and Byron C Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the Conference. Association for Computational Linguistics. Meeting*, volume 2023, page 15566.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023b. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*. Version 4.

Siyuan Wang, Zhongkun Liu, Wanjun Zhong, Ming Zhou, Zhongyu Wei, Zhumin Chen, and Nan Duan. 2022. From lsat: The progress and challenges of complex reasoning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2201–2216.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Yilin Wen, Zifeng Wang, and Jimeng Sun. 2024. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10370–10388.

Junda Wu, Tong Yu, Xiang Chen, Haoliang Wang, Ryan Rossi, Sungchul Kim, Anup Rao, and Julian McAuley. 2024. Decot: Debiasing chain-of-thought for knowledge-intensive tasks in large language models via causal intervention. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14073–14087.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models. *arXiv preprint arXiv:2309.11674*. Version 2.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, and Jian-Guang Lou. 2024. Re-reading improves reasoning in language models.

Junjie Ye, Nuo Xu, Yikun Wang, Jie Zhou, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llm-da: Data augmentation via large language models for few-shot named entity recognition. *arXiv preprint arXiv:2402.14568*.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations*.

Hang Zhang, Yeyun Gong, Yelong Shen, Weisheng Li, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2021. Poolingformer: Long document modeling with pooling attention. In *International Conference on Machine Learning*, pages 12437–12446. PMLR.

Kai Zhang, Bernal Jiménez Gutiérrez, and Yu Su. 2023a. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023b. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Ziqi Zhang, Cunxiang Wang, Xiao Xiong, Yue Zhang, and Donglin Wang. 2024. Nash cot: Multi-path inference with preference equilibrium. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14572–14587.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. 2024. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Computing Surveys*, 56(11):1–39.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781.

A Algorithms

The process of Entity Selection in Algorithms 1, Path Generation in Algorithms 2.

Algorithm 1 Entity Selection

Input: Subgraph $G_s = (E, R)$, query q, context x, embedding model M
Output: Key entities E_{key}, starting entity e_s

$E_{\text{key}} \leftarrow \emptyset$
for $e \in E$ **do**
  $s_{\text{sem}} \leftarrow \frac{\mathbf{v}_e \cdot \mathbf{v}_q}{\|\mathbf{v}_e\| \|\mathbf{v}_q\|}$
  $s_{\text{key}} \leftarrow s_{\text{sem}} + w_q \mathbf{I}_q(e) + w_x \mathbf{I}_x(e)$
  $s_{\text{start}} \leftarrow \frac{d_{\text{out}}(e)/d_{\max} + s_{\text{sem}}}{2}$
end for
$E_{\text{key}} \leftarrow \text{top}_k\{s_{\text{key}}(e) \mid e \in E\}$
$e_s \leftarrow \arg\max\{s_{\text{start}}(e) \mid e \in E\}$
return E_{key}, e_s

Algorithm 2 Path Generation

Input: Subgraph $G_s = (E, R)$, query q, context x, key entities E_{key}, starting entity e_s, embedding model M, LLM
Output: Path set P

$P \leftarrow \emptyset$, $h_{\max} \leftarrow 5$
for (e_i, e_{i+1}) in pairs(E_{key}) **do**
  if has_path(G_s, e_i, e_{i+1}) **then**
    $P \leftarrow P \cup \{\text{BFS}(e_i, e_{i+1}) \mid l(p) \leq h_{\max}\}$
  end if
end for
for $e_t \in E \setminus \{e_s\}$ **do**
  if has_path(G_s, e_s, e_t) **then**
    $P \leftarrow P \cup \{\text{BFS}(e_s, e_t) \mid l(p) \leq h_{\max}\}$
  end if
end for
for $p \in P$ **do**
  if $s_{\text{sem}}(p, q) > \theta_{\text{sem}}$ and $l(p) < h_{\max}$ **then**
    $P \leftarrow P \cup \text{extend}(p, G_s, q, \alpha)$
  end if
end for
$P \leftarrow P \cup LLM(e_s, E, R, q, x, h_{\max})$
return P

B Baselines

Vanilla LLM, employs in-context learning to directly predict answers by presenting tasks and questions without intermediate reasoning steps.

CoT (Wei et al., 2022), generates step-by-step explanations to derive answers, enhancing reasoning through structured intermediate steps.

CoT-SC (Wang et al., 2023c), samples multiple CoT reasoning paths and selects the most frequent answer via majority voting to improve robustness.

Auto-CoT (Zhang et al., 2023b), automatically constructs multi-step reasoning sequences in natural language, reducing the need for manual prompt design.

Complex-CoT (Fu et al., 2023), adopts a complexity-based approach, sampling multiple CoT paths and choosing answers that align consistently across complex reasoning chains through voting.

PS and PS+ (Wang et al., 2023a), utilize zero-shot CoT by dividing tasks into planning and solving phases to generate answers. PS+ incorporates additional details, such as variables, to facilitate the reasoning process.

RE2 (Xu et al., 2024), enhances reasoning by rephrasing and re-reading the question before generating CoT steps, serving as a plug-and-play method.

ERA-CoT (Liu et al., 2024), the current SOTA, captures relations between entities and supports reasoning across diverse tasks through CoT, leveraging structured entity interactions for improved performance.

Dataset	Question Type	Num.	Domain
CommonsenseQA	multi-choice	3741	Commonsense Reasoning
StrategyQA	multi-choice	1580	Commonsense Reasoning
ARC-Challenge	multi-choice	1695	Commonsense Reasoning
Date Understanding	multi-choice	250	Symbolic Reasoning
Object Tracking	multi-choice	250	Symbolic Reasoning
Last Letters	question-answering	500	Symbolic Reasoning
LogiQA	multi-choice	3688	Logical Reasoning
ReClor	multi-choice	2069	Logical Reasoning
AR-LSAT	multi-choice	1523	Logical Reasoning
AQuA	multi-choice	3850	Arithmetic Reasoning
GSM8K	number words	3500	Arithmetic Reasoning
SVAMP	number words	1000	Arithmetic Reasoning

Table 4: Dataset statistics, where "Num." represents the number of sampled datasets.

C Dataset Statistics

Table 4 provides detailed information about the data included in the experiment, where the sampled data are randomly selected from datasets.

D Parameter Settings

For entity and relation extraction, we employ an LLM in a zero-shot setting, generating $n_p = 5$ reasoning paths per task. Implicit relations are scored by LLM, retaining those with confidence scores above a threshold $\theta_r = 0.7$. For multi-hop path generation, we retain up to $n_{\text{path}} = 5$ paths via Beam Search, filtering paths with a semantic similarity threshold $\theta_{\text{sem}} = 0.5$ (computed using all-MiniLM-L6-v2) and extending them with a factor $\alpha = 0.5$. The generation temperature is set to 0.3 to ensure stable outputs. In question answering, answers are aggregated via majority voting with a threshold $\tau = 2$, iterating up to $n_{\text{attempt}} = 3$ times if needed.

E Evaluation Metrics

We use accuracy and exact match as the evaluation metric for different datasets. Specifically, for datasets like CommonsenseQA, AR-LSAT, and Object Tracking that contain options, we utilize the accuracy based on whether the options match the standard answers. For problems like SVAMP, where the output is a number, we use regular expressions for exact match judgment of the answers. For datasets like Last Letter that do not contain question options, we compare the output with answer alternatives and also use the exact match method for accuracy estimation. The same processing approach is adopted for different methods across these datasets.

F Example

Table 5 shows an example procedure of the DS-MHP on AR-LSAT dataset.

Context: Each of five illnesses—J, K, L, M, and N—is characterized by at least one of the following three symptoms: fever, headache, and sneezing. None of the illnesses has any symptom that is not one of these three. Illness J is characterized by headache and sneezing. Illnesses J and K have no symptoms in common. Illnesses J and L have at least one symptom in common. Illness L has a greater number of symptoms than illness K. Illnesses L and N have no common symptoms. Illness M has more symptoms than illness J.
Question: If Walter has exactly two of the three symptoms, then he cannot have all of the symptoms of ?
Choices: A: both illness J and illness L, B: both illness J and illness N, C: both illness K and illness L,
D: both illness K and illness N, E: both illness L and illness N
Answer: E

Dynamic Subgraph Construction
NER: [j, k, l, m, n, fever, headache, sneezing, walter]
Relation Extraction:
Explicit Relation:
[(j, has_symptom, headache), (j, has_symptom, sneezing), (j, no_common_symptom, k), (j, common_symptom, l), (l, more_symptoms, k), (l, no_common_symptom, n), (m, more_symptoms, j)]
Implicit Relations:
[(k, has_symptom, fever), (l, has_symptom, fever), (l, has_symptom, headache), (m, has_symptom, fever), (m, has_symptom, headache), (m, has_symptom, sneezing), (n, has_symptom, sneezing)]
Relation Assessment:
(k, has_symptom, fever): 0.90
(l, has_symptom, fever): 0.85
(l, has_symptom, headache): 0.95
(m, has_symptom, fever): 0.90
(m, has_symptom, headache): 0.90
(m, has_symptom, sneezing): 0.90
(n, has_symptom, sneezing): 0.85
Final Relations:
[(j, has_symptom, headache), (j, has_symptom, sneezing), (j, no_common_symptom, k), (j, common_symptom, l), (l, more_symptoms, k), (l, no_common_symptom, n), (m, more_symptoms, j), (k, has_symptom, fever), (l, has_symptom, fever), (l, has_symptom, headache), (m, has_symptom, fever), (m, has_symptom, headache), (m, has_symptom, sneezing), (n, has_symptom, sneezing)]
Subgraph:
j -has_symptom-> headache, j -has_symptom-> sneezing, j -no_common_symptom-> k, j -common_symptom-> l, l -more_symptoms-> k, l -no_common_symptom-> n, m -more_symptoms-> j, k -has_symptom-> fever, l -has_symptom-> fever, l -has_symptom-> headache, m -has_symptom-> fever, m -has_symptom-> headache, m -has_symptom-> sneezing, n -has_symptom-> sneezing

Multi-Hop Path Generation
Entity Selection:
Key entities: [walter, j, l, n]
Start entity: walter
Path Generation:
Key Entity Joint Paths: j -common_symptom-> l -no_common_symptom-> n
Starting Entity Paths: walter has no outgoing edges, so no BFS paths.
LLM-Supplemented Paths: walter -> headache -> j, walter -> fever -> l, walter -> sneezing -> n,
Discarded (no edges from walter).
Path Refinement:
j -common_symptom-> l -no_common_symptom-> n

Question Answering
Answer Generation: [E, E, E, E, E]
Majority Voting: E
Answer: E

Table 5: Case On AR-LSAT.