Structured Knowledge Representation for Reasoning in Deep Models

A thesis submitted in fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Analytics

by

Sirui Huang

to

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney

NSW - 2007, Australia

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Sirui Huang* declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science* at the *Faculty of Engineering and Information Technology* at the University of Technology Sydney, Australia. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with the *Hong Kong Polytechnic University*.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

DATE: 23th June, 2025

PLACE: Sydney, Australia

Abstract

Structured knowledge—represented through tabular formats, bipartite graphs, and knowledge graphs—pervades real-world systems as a foundational organizational framework for critical decision-making processes in domains such as laboratory documentation, financial reporting. These structures enable explicit organization of knowledge while encapsulating insights through patterns and relationships unattainable via unstructured data alone. Despite their operational value, the construction of such knowledge structures remains resource-intensive, necessitating specialized domain expertise and enterprise-level system integrations. Consequently, optimizing the utilization of these meticulously organized knowledge repositories becomes imperative. While deep learning demonstrates proficiency in raw data processing, its capacity for symbolic reasoning over structured knowledge remains constrained by inherent challenges: reconciling discrete relational logic with neural architectures, addressing sparse connectivity, and maintaining interpretability.

This thesis systematically addresses these limitations through a unified methodological framework that advances reasoning capabilities across structured knowledge systems, encompassing innovations from data-level infrastructure to representation-level architectures. At the data-level, we establish a unified benchmarking protocol to examine distinctive structural properties while enabling comprehensive analytical capabilities. From representation-level, we engineer three novel computational models that resolve critical challenges including static heterogeneous dependency management and dynamic adaptation mechanisms within structured knowledge systems. By innovatively synthesizing human-curated knowledge with data-driven learning paradigms, this research advances artificial intelligence systems capable of harnessing structured knowledge's full potential, thereby mediating the historical dichotomy between symbolic reasoning and neural learning approaches.

Acknowledgments

I would like to express my deepest gratitude to my supervisors, Prof. Guandong Xu and Prof. Qing Li for their patient and steadfast guidance throughout my doctoral journey. Their mentorship has profoundly shaped both my research skills and academic perspective, leaving an indelible impact on my future endeavors. During my time at The Hong Kong Polytechnic University, Prof. Qing Li constantly encouraged me to identify meaningful research questions with innovative techniques. At the University Technology Sydney, Prof. Guandong Xu helped me refine my technical expertise and academic writing. Their guidance has been instrumental in my growth as an independent researcher. I am truly fortunate to have had the opportunity to learn from Prof. Qing Li and Prof. Guandong Xu during this pivotal stage of my academic career, and I will always be grateful for their invaluable support.

I would like to express my sincere appreciation to Dr. Qian Li for guiding me in exploring the intricacies of causality. She has an exceptional ability to help me organize my disordered ideas, turning them into clear insights. Her profound theoretical knowledge and meticulous attitude have consistently inspired and influenced my academic journey.

I also owe Dr. Xuming Hu my heartfelt thanks for his mentoring in the field of large language models. His profound expertise, insightful perspectives, and unwavering support have been a constant source of motivations, lifting me during my darkest moments.

I am deeply grateful to my co-supervisors at UTS, Dr. Xianzhi Wang, Dr. Angela Huo and Dr. Dayong Ye. Their unwavering support and guidance have been instrumental in my doctoral journey.

Moreover, I would like to thank all my colleagues for their support in both my research and daily life, as well as their invaluable feedback and suggestion to my research works: Yicong Li, Haoran Yang, Changmeng Zheng, Shijie Wang, Lin Wang, Jing Long, Xueyao Sun, Xiangmeng Wang, Dianer Yu, Dawei Xu, Jingran Su, Yunqing Liu, Kaize Shi, Zhangkai Wu, Jin Li, Zhuo Cai, Yanggan Gu, Hanqian Li, and all my group mates.

Finally, I would like to express my heartfelt gratitude to my parents, my brother, my sister-in-law. My family has been a constant source of support, navigating me overcame every challenge along this long journey.

List of Publications

Related to the Thesis :

1. **Sirui Huang**, Qian Li, Haoran Yang, Dianer Yu, Qing Li, Guandong Xu. Causal Time-aware News Recommendations with Large Language Models, *ACM Transactions on Information Systems (TOIS)*, 2025. [ERA&CORE: A*, JCR Q1]

2. **Sirui Huang***, Hanqian Li*, Yanggan Gu, Xuming Hu, Qing Li, Guandong Xu. HyperG: Hypergraph-Enhanced LLMs for Structured Knowledge, in *The 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2025. [ERA&CORE: A*]

3. **Sirui Huang***, Yanggan Gu*, Zhonghao Li, Xuming Hu, Qing Li, Guandong Xu. StructFact: Reasoning Factual Knowledge from Structured Data with Large Language Models. in *The 63rd Annual Meeting of the Association for Computational Linguistics (ACL), 2025*. [ERA&CORE: A*]

4. **Sirui Huang**, Qian Li, Xiangmeng Wang, Dianer Yu, Guandong Xu, Qing Li. Counterfactual Debasing for Multi-behavior Recommendations, *International Conference on Database Systems for Advanced Applications (DASFAA)*, 2024. [ERA&CORE: B]

Others :

6. Yicong Li*, **Sirui Huang***, Yu Yang, Haoran Yang, Qing Li, Jiannong Cao, Kwok Shing CHENG, Guandong Xu. Time-aware Hypergraphs Neural Network for Explainable Sequential Recommendation.Under Review in *ACM Transactions on Information Systems (TOIS)*. [ERA&CORE: A*, JCR Q1] (Submitted ID: TOIS-2024-0681)

7. Haoran Yang, Xiangyu Zhao, **Sirui Huang**, Shanshan Ye, Guandong Xu. Latex-gcl: Large language models (llms)-based data augmentation for text-attributed graph contrastive learning, *arXiv preprint arXiv:2403.11122*, 2024.

8. Dianer Yu, Qian Li, **Sirui Huang**, Guandong Xu. Large Language Models meet Causal Inference: Semantic-Rich Dual Propensity Score for Unbiased Sequential Recommendation. Under Review in *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. [ERA&CORE: A*, JCR Q1] (Submission ID: TKDE-2024-12-3260)

9. Lin Wang, Shijie Wang, **Sirui Huang**, Qing Li. Simplifying Graph Neural Kernels: from Stacking Layers to Collapsed Structure. Under Review in *The 42th International Conference on Machine Learning (ICML 2025)*. [ERA&CORE: A*] (Submission ID: 1174)

10. Yanggan Gu, Junzhuo Li, **Sirui Huang**, Xin Zou, Zhenghua Li, Xuming Hu. Capturing Nuanced Preferences: Preference-Aligned Distillation for Small Language Models. Under Review in *The 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*. [ERA&CORE: A*] (Submission ID: 7650)

11. Yunkai Dang, Kaichen Huang, Jiahao Huo, Yibo Yan, **Sirui Huang**, Dongrui Liu, Mengxi Gao, Jie Zhang, Chen Qian, Kun Wang, Yong Liu, Jing Shao, Hui Xiong, Xuming Hu. Explainable and interpretable multimodal large language models: A comprehensive survey, *arXiv preprint arXiv:2412.02104*.

12. Zhonghao Li, Xuming Hu, Aiwei Liu, Kening Zheng, **Sirui Huang**, Hui Xiong. Refiner: Restructure retrieval content efficiently to advance question-answering capabilities. *Findings of the Association for Computational Linguistics (EMNLP 2024)*. [ERA&CORE: A*]

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

In this chapter, we briefly introduce the background and definition of structured knowledge, outline related challenges and research questions, present the main contributions of the thesis, and conclude with an overview of the thesis framework.

1.1 Background

Structured knowledge is information that is systematically organized into predefined formats, making it easy to use for various downstream tasks. For example, a patient's laboratory sheet with labeled columns for test name, result, and reference range, or a bipartite graph showing interactions between users and items. In contrast, unstructured data, such as a customer's email complaint or a social media post, lacks a fixed format and exists in flexible forms. The value of unstructured data lies in its context but requires further interpretation using tools like natural language processing or AI models to uncover patterns. The core difference lies in organization: structured knowledge follows an explicit structure with logical dependencies, whereas unstructured knowledge relies on additional interpretation.

Beyond its well-organized structure, structured knowledge exhibits high usability by seamlessly integrating into various downstream tasks. For example, in Figure 6.1 when purchasing a new iPhone, Apple.com provides a structured comparison table that organizes key specifications of different products, helping users make informed decisions. Also, this structured table also help AI models to learn the differences among products

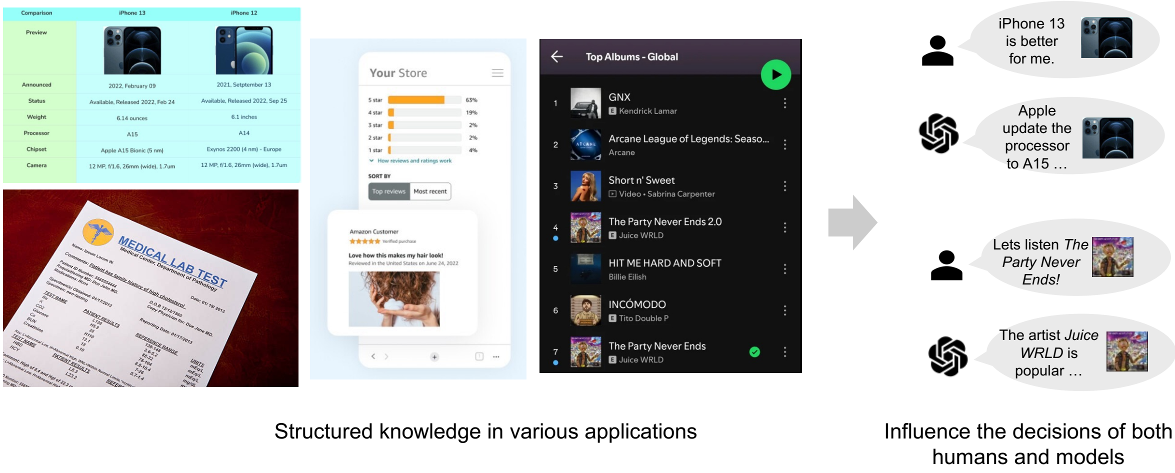


Figure 1.1: Examples of structured knowledge in real life and how it influences the decisions of both humans and deep models.

and provide better recommendations and suggestions to users. An another example of real-world structured knowledge is in Spotify. Structured playlists and ranking data enhance music discovery, allowing users to find the music they may like and guiding AI models to identify trending artists. Structured knowledge powers product reviews and rating aggregation, enabling more personalized recommendations. Even in some high-stake scenarios such as healthcare, structured lab reports provide critical insights for medical diagnoses. By influencing both human and AI-driven decisions, structured knowledge underscores its versatility and practical value across multiple domains.

Despite its highly organized nature and high usability, establishing structured knowledge comes with a significant cost due to the specialized processing required to filter, clean, and organize raw data into structured formats. This involves not only sophisticated natural language processing and data science techniques but also expert knowledge and robust warehousing infrastructure. For instance, the U.S. Congress has provided financial incentives of up to $44,000 per physician to support the establishment of electronic health records (EHR) [152], a structured repository essential for medical analysis and decision support. Additionally, storing and managing structured data requires costly systems with strict schemas, such as MongoDB and Oracle. According to a LinkedIn report [80], the global warehousing and storage market was valued at $504 billion in 2023 and is expected to more than double by 2030. Ensuring data consistency, updating structured repositories, and maintaining organized storage add to the ongoing expenses, making structured knowledge a resource-intensive but invaluable asset.

In the era of AI, to fully utilize these precious organized knowledge in various downstream tasks, we investigate the reasoning process in both traditional machine learning models and deep learning models. Traditional models, such as decision trees, reason over structured knowledge through hierarchical rule-based splitting, where features are evaluated at each node to make deterministic decisions. For example, a tree model classifying medical conditions might first check symptoms such as fever and cough, then further split based on test results to reach a diagnosis. This approach relies on explicit, interpretable rules but struggles with capturing complex and high-dimensional relationships among data. In contrast, deep learning models, reason over structured knowledge by learning implicit correlations and modeling latent representations. These models capture nuanced dependencies through latent representations, enabling richer contextual understanding. Graph Neural Network (GNN) is a typical deep learning model that is designed for modeling structures. For example, the interactions between users and items can be built as connections between nodes in a graph and a GNN is applied to infer the probability of users' next item interaction from not only the node features but also the topological connections. Although deep models excel at capturing complex relationships, their reasoning processes are less transparent compared to those of traditional machine learning approaches. This raises important questions about whether structured knowledge embedded in organized data is being effectively leveraged, and how reasoning is performed within these models.

This thesis investigates the use of structured knowledge in deep models, spanning from the data level to the representation level. The following section in this chapter elaborates on the motivations in detail and summarizes the main contributions.

1.2 Motivation

Structured knowledge is both ubiquitous and essential across a wide range of downstream applications, providing rich, organized information that supports informed decision-making. Deep models have demonstrated impressive effectiveness across various domains, owing to their strong ability to capture nuanced correlations through latent representations. However, the implicit learning of structured knowledge in deep models raises concerns about whether such valuable information is being utilized effectively. This observation motivates us to systematically evaluate and improve the reasoning capabilities of deep models when applied to structured data.

Data are at the forefront of deep learning, driving the need for a sophisticated exploration of structured knowledge. Although deep models have achieved remarkable success in processing unstructured data (e.g., emails or customer reviews), they often face limitations when it comes to effectively leveraging the structured, relational information embedded in structured repositories—unless explicitly adapted through techniques like feature engineering, graph modeling, or tabular learning. Even highly organized structured data must undergo feature engineering or augmentation before being fed into deep models. This underscores the importance of carefully processing the unique characteristics of structured knowledge, which promises to enhance data interpretation, improve prediction accuracy, and drive more reliable decision-making in practical applications.

From the perspective of the model, learning effective latent representations is essential for enabling deep models to reason over structured knowledge. Deep models learn the parameters of hidden layers from extensive training data, capturing many subtle correlations, some of which may even imperceptible to humans. In the inference stage, the model relies on these latent representations to implicitly model structured knowledge within the learned representation space. Unlike the modeling of unstructured text, learning representations for structured knowledge requires more than capturing local features or pairwise relationships between entities. It also demands understanding and modeling complex, irregular, and even heterogeneous topological structures—such as semantic dependencies across tree-like, graph-based, or multi-hop paths. For example, in recommendation systems, where user-item interactions are not only multi-relational (e.g., click, purchase, view) but also dynamic—evolving over time based on user preferences and contextual factors. Effectively modeling such relation heterogeneity and temporal dynamics requires that representations adapt to shifting interaction patterns and capture the semantics of diverse relation types across time.

In summary, while deep models have demonstrated impressive performance, particularly on unstructured data, there is still considerable room to improve their ability to reason over structured knowledge. Since structured knowledge is prevalent across many domains, addressing this gap is important for enhancing the effectiveness of a wide range of downstream applications. This challenge exists at both the data level and the representation level. At the data level, structured knowledge carries unique characteristics, such as explicit relationships, hierarchical organization, and semantic consistency. At the representation level, it is essential to develop approaches that help deep models effectively encode and learn latent representations that reflect these structures. Progress in both areas is crucial for enabling deep models to perform well in decision-making tasks that rely on structured reasoning.

1.3 Research Questions

To explore the reasoning capability of deep models over structured knowledge, we propose the following research questions, spanning the data level (RQ1) and the representation level (RQ2.1, RQ2.2 and RQ3).

1.3.1 Data-level

At the data level, we examine the reasoning capabilities of deep models from the perspective of the unique characteristics of structured knowledge.

RESEARCH QUESTION 1 (RQ1): How do the unique data characteristics of structured knowledge challenge reasoning?

Unlike unstructured data, structured knowledge features unique characteristics such as organized dependencies and hierarchies that need extra processing such as feature engineering to handle effectively by neural networks. These characteristics require models to understand and process not just correlations, but also the intricacies topology embedded in the logical and semantic organization. For example, in a patient's laboratory sheets, the model must integrate and infer relationships between the results of different tests and their corresponding reference ranges. This necessitates a precise understanding of topological dependencies and the heterogeneity of different data types (e.g., numeric and textual data).

1.3.2 Representation-level

At the representation level, we aim to enhance representation learning by addressing key challenges in both static and dynamic scenarios, respectively: (1) heterogeneous dependencies representations (RQ2.1, RQ2.2), and (2) dynamic adaptation to evolving dependencies and entity representations (RQ3).

RESEARCH QUESTION 2.1 (RQ2.1): How can spurious correlations be debiased in representations of heterogeneous dependencies?

Compared to homogeneous dependencies, though different types of relations in structured knowledge introduce additional information that can enhance representation learning, these heterogeneous dependencies also bring a significant challenge: they tend to introduce noise into the latent representations through the presence of more spurious correlations. Such noise can obscure the true underlying patterns and relationships that are critical for accurate predictions by deep models. This issue is common in a wide range of applications, such as the multi-behavior interactions between users and items in recommender systems, where spurious correlations can lead to faulty recommendations and diminished user satisfaction.

RESEARCH QUESTION 2.2 (RQ2.2): How can heterogeneous dependencies and unstructured data be effectively integrated in representation?

Another problem concerning heterogeneous dependencies in static scenarios is the difficulty of combining it with unstructured data. Given the vast amount of unstructured data, deep models must be capable of seamlessly integrating structured knowledge with unstructured data to enable wide-ranging applications. For example, Large Language Models (LLMs) need to understand unstructured textual queries from users and recall and reason over the structured data fed during pretraining to answer these queries.

RESEARCH QUESTION 3 (RQ3): How to adapt structured knowledge representation to temporal and contextual evolutions?

Beyond static scenarios, the dynamic evolutions of either temporal or contextual information are expected to be encoded in representations. This dynamic question is essential because the relevance and accuracy of structured knowledge can change over time and in different contexts. For example, in news recommendations, the relevance of a news item to a particular user might diminish in future recommendations as its freshness decreases, but it could also potentially increase if another news item related to the same context gains popularity. By enhancing the deep models to dynamically adapt their knowledge representations, we can improve their applicability and accuracy in the evolving real-world environments.

1.4 Research Innovations

This thesis aims to improve structured knowledge reasoning in deep models through two main steps: (1) evaluating the reasoning capabilities of state-of-the-art deep models over structured knowledge, and (2) enhancing representation learning from static to dynamic scenarios. The main innovations of this study are summarized as follows:

Innovation 1. We introduce a unified benchmark designed to facilitate in-depth analyses of structured knowledge through the lens of unique data-level characteristics, providing comprehensive and meticulous annotations of five novel challenges on both existing and newly collected structured knowledge sources, along with their corresponding questions. Extensive experiments and analyses on ten models, varying in parameter sizes and training strategies, uncover several common limitations and provide valuable insights for future improvements.

Innovation 2. We alleviate the bias introduced by spurious correlations among heterogeneous dependencies by incorporating a constraint based on learnable counterfactual examples during training. This approach can be better understood from a causal perspective. Taking different types of user behaviors in recommendation systems as an example of heterogeneous dependencies in structured interaction data, our method limits the confounding effects arising from multiple interaction types. As a result, it improves ranking performance and enables more accurate recommendations.

Innovation 3. To effectively integrate unstructured data with structured knowledge in latent representations, we propose a novel hypergraph-based framework that accurately encodes structured knowledge and seamlessly incorporates it with unstructured queries expressed in natural language. This integration is primarily achieved through a newly designed hypergraph neural network, capable of capturing and unifying the semantics of both structured and unstructured information.

Innovation 4. To address the dynamic challenges posed by temporal evolution over structured data and contextual evolution over unstructured textual information, we propose a novel model that takes advantage of the complementary strengths of different deep models to capture both time-variant and time-invariant representations. Our proposed model enhances the expressiveness of latent representations in scenarios such as news recommendation, where prediction accuracy is highly sensitive to evolutions in both temporal patterns and contextual information.

1.5 Research Contributions

The key contributions of the research presented in this thesis are summarized as follows.

Contribution 1. Through explorations with our proposed benchmark StructFact, we evaluate how the understanding and reasoning over structured knowledge performs across task signified by different data characteristics. For example, in tasks that rely on heterogeneous data, deep models represented by LLMs heavily depend on the order of information for understanding. We aim for StructFact to serve as a compass in exploring the boundaries of LLMs in knowledge-sensitive tasks involving structural facts, while also advancing their practical applications in real-world scenarios.

Contribution 2. This study introduces several models aimed at enhancing representations to address the various challenges outlined in the previously stated research questions.

1) An counterfactual debiased multi-behavior recommender system (C-MBR) is developed, leveraging the ordering inherent in observational data and counterfactual examples to guide and constrain the reasoning process.

2) HyperG, a novel structured knowledge reasoning framework, is built upon a novel hypergraph neural network to effectively combine structured and unstructured knowledge at the representation level.

3) CAST-Rec, a model that utilizes a causally-enhanced Transformer architecture with the capabilities of large language models (LLMs) to effectively capture the dynamic evolution of temporal and contextual patterns in news recommendation tasks.

Contribution 3. This research systematically investigates and addresses the challenges of structured knowledge reasoning, spanning from data-level characteristics to representation learning.

1) Data level: StructFact benchmark, which identifies six key data characteristics and defines five corresponding tasks to enable comprehensive, data-driven analysis.

2) Heterogeneous dependencies representation: C-MBR and HyperG address the complexities of heterogeneous dependencies within different types of structured knowledge.

3) Dynamic evolution representation: CAST-Rec captures both temporal and contextual evolutions by integrating representations from diverse deep learning models.

1.6 Research Significance

The significance of this thesis lies in its practical relevance. Structured knowledge reasoning plays a vital role in real-world domains where structured data is central to decision-making and analysis. For example, patients' laboratory results and medical histories are commonly stored in electronic health records (EHRs) using organized data formats, while financial statements and audit documents are often presented in tables. These structured formats not only convey critical information efficiently but also form the foundation for more advanced analytics and decision-making.

In the era of AI, increasing numbers of disciplines rely on deep learning models to uncover latent patterns within structured datasets. This research contributes to that progression by developing more effective representation learning techniques for structured knowledge reasoning. By overcoming the limitations of unstructured or heuristic-based approaches, our models enable more robust and accurate predictions to support data-driven decision-making. For example, the proposed hypergraph learning framework

(Chapter 5), HyperG, enhances the reasoning capability of LLMs on structured knowledge, thereby promoting broader applications of language models to structured EHRs for more accurate disease diagnosis. Furthermore, integrating causal reasoning with deep models enhances their capabilities that are especially important in high-stakes scenarios like healthcare and finance. For example, with the newly proposed CAST-Rec model (Chapter 6), the developers can analyze the rationale behind predictions and provide users with more transparent recommendations.

We argue that structured knowledge reasoning is more than a technical challenge; it is a transformative process that converts raw data into actionable insights. Its practical impact lies in equipping AI systems with the ability to reason effectively with structured information, thereby amplifying their utility across a wide range of real-world applications.

1.7 Thesis Structure

We depict the structure of this thesis in Figure 1.2, and organize the content of each chapter as follows.

CHAPTER 1 provides an introduction to the research background, motivation, research questions, contributions, and significance of this study.

CHAPTER 2 reviews related work on structured knowledge reasoning, with a particular focus on user-item interaction modeling in recommender systems and tabular data reasoning. It discusses existing deep learning models, applications, and limitations in these domains.

CHAPTER 3 proposes benchmark StructFact, investigating **RQ1** by extensive annotations and in-depth analyses from the perspectives of five designed tasks based on the unique charateristics of structured knowledge.

CHAPTER 4 proposes C-MBR, a counterfactual-enhanced multi-behavior recommendation model. This chapter addresses **RQ2.1** by debiasing the noise in multi-behavior modeling with stable counterfactual examples and reasoning.

CHAPTER 5 introduces HyperG: a hypergraph-based framework combining the unstructured and structured knowledge via semantic-aware message passing and aggregation in hypergraph neural networks, addressing **RQ2.2**.

CHAPTER 6 presents CAST-Rec, moving from the static scenario to dynamic adaptation, capturing evolutions of both time and contexts by integrating the representations of

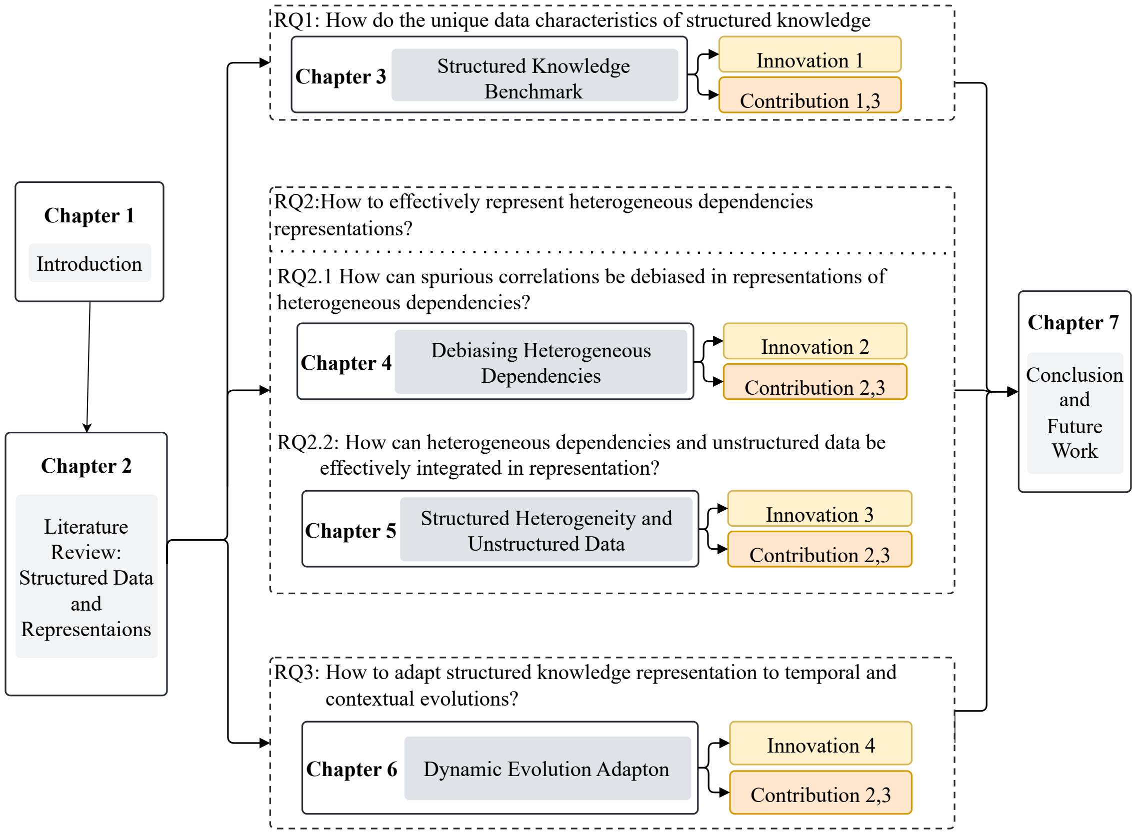


Figure 1.2: Overview of thesis structure.

LLMs and transformer, yielding improved performance in news recommendations. This work carefully addresses **RQ3**.

CHAPTER 7 summarizes the contributions of past research works, discusses the corresponding implications and limitations, and proposes future directions.

CHAPTER 2

Literature Review

Structured knowledge reasoning has obtained significant attention across several application domains such as recommender systems, tabular understanding, and healthcare analytics, etc. This section reviews existing approaches to structured knowledge reasoning, examined through the lenses of structured data and representation methodologies.

2.1 Structured Data

Structured data, often organized in formats such as tables, graphs, and lists, is distinguished from unstructured data like free-form text by its well-defined schema and explicit internal relationships. Reasoning over such data requires models to consider not only the content itself but also the underlying structure and relational dependencies. From a data perspective, a wide range of benchmarks have been developed to extract structured knowledge from real-world sources and evaluate the reasoning capabilities of deep models. In this section, we review existing benchmarks and empirical studies across different types of structured knowledge and summarize commonly used datasets for various tasks in Table 2.1.

2.1.1 Tabular and Relational Data

Early work in structured reasoning centered on rule-based information systems and symbolic logic applied to relational databases, enabling deeper analysis and inference.

Table 2.1: A comprehensive comparison of various benchmarks with structured knowledge.

Dataset	Task	Source	Evidence/Data Type	Answer Type	Domain
ToTTo [99]	Generation	Wikipedia	Table		General
TaKG [105]	Generation	Wikipedia	Table, Graphs, Text		General
WebNLG [36]	Generation	DBPedia	Graphs		General
DART [95]	Generation	Wikipedia	Table, Graphs		General
LOTNLG [182]	Generation	Wikipedia	Table		General
SQA [56]	Question Answering	Wikipedia	Table	Span	General
NQ-tables [46]	Question Answering	Wikipedia	Table	Span	General
HybridQA [23]	Question Answering	Wikipedia	Table, Text	Span	General
WikiTableQuestion(WTQ) [100]	Question Answering	Wikipedia	Table, Text	Span	General
FetaQA [94]	Question Answering	Wikipedia	Table, Text	Span	General
TAT-QA [188]	Question Answering	Wikipedia	Table, Text	Span	General
Open-WikiTable [66]	Question Answering	Wikipedia	Table, Text, SQL	Span	General
WebSRC [24]	Question Answering	Web pages	HTML	Span/Boolean	General
OTTQA [21]	Question Answering	Wikipedia	Table, Text	Multiple Choice	General
TSQA [73]	Question Answering	Exam	Table	Multiple Choice	Geography
ROBUT [183]	Question Answering	WTQ, WIKISQL-WEAK [186], and SQA	Table	Span	General
SUC [120]	Question Answering	Wikipedia	Table	Span	General
MiMoTable [78]	Question Answering	Baidu Wenku, Google Search	Table	Span	General
TableBench [157]	Question Answering	Wikipedia	Table	Span	General
DocTabQA [139]	Question Answering	Securities and Exchange Commission	Table	Span	Finance
FREB-TQA [187]	Question Answering	WTQ,WikiSQL,SQA,TAT [188]	Table	Span	General
SciTabQA [37]	Question Answering	SciGen [93]	Table	Span	General
GraphWiz [18]	Question Answering	Self-developed	Graph	Span	N/A
GLBench [77]	Question Answering	citation networks, web links, and social networks	Graph	Span	General
FEVEROUS [3]	Fact-checking	Wikipedia	Table, List, Text	Fact/Non-Fact/NEI	General
TabFact [22]	Fact-checking	Wikipedia	Table	Fact/Non-Fact	General
Infotabs [40]	Fact-checking	Wikipedia	Table	Fact/Non-Fact/NEI	General
Fact-KG [61]	Fact-checking	WebNLG [36],DBPedia	Graphs	Fact/Non-Fact	General
Semeval 2021 Task 9 [142]	Fact-checking	Scientific Articles	Table, Text	Fact/Non-Fact/NEI	Science
PubHealthTab [2]	Fact-checking	PubHealth [64], Wikipedia	HTML	Fact/Non-Fact	Healthcare
WikiSQL [**?**]	Text-to-SQL	Wikipedia	Table	code	General
Spider [174]	Text-to-SQL	College database, SQL tutorial websites, online csv files, DatabaseAnswers[a], WikiSQL	Table	code	General
BIRD [71]	Text-to-SQL	Kaggle, CTU Prague Relational Learning Repository	Table	code	General
ARCADE [172]	Text-to-SQL	GitHub	Table	code	General

[a] http://www.databaseanswers.org/

Among various structured formats, tabular data are the most prevalent and play a vital role in high-stakes applications [13]. Several benchmarks have been developed to capture structured knowledge from public sources and annotate them for different tasks. Table Question Answering (TQA), Table-to-Text, Text-to-SQL, are three common tasks based on the understanding and reasoning over tabular data. For example, WikiTableQuestion (WikiTQ) [100] and TabFact [22] extract tables from WiKipedia pages and wrote questions targetting specific cells or protions of those tables, while ToTTo [99] focuses on the understanding and generation ability of deep language models. We summarize the benchmarks that contain tabular knowledge in Table 2.1, organized by different targeted tasks. Later, some benchmarks are motivated by particular challenges in tabular reasoning. For example, ARACADE [172] and BIRD [71] focus on the text-to-SQL task over super long table.

Building on these benchmarks, pioneering researchers have conducted empirical studies to evaluate the performance of deep models on tabular data. Early studies compared deep learning models (e.g., neural networks) with traditional machine learning approaches (e.g., gradient-boosted decision trees) and found the latter to outperform deep models in many tabular scenarios [12, 116]. Subsequent work comparing various neural architectures revealed that transformer-based models [136] often underperform compared to more conventional sequential models such as RNNs when applied to tabular data [176]. These findings raise questions about whether language models, which are predominantly transformer-based, can effectively handle tabular tasks as successfully as they do in unstructured domains like text style transformation. With the advert of Large Language Models (LLMs), recent empirical studies on tabular data focus on evaluating the related capability of LLM. Form the perspective of instruction-tuning, researchers investigates different prompting strategies in understanding [39, 81, 123] and question answering [9, 85, 105**?**]. For example, [85] find that integrating textual and symbolic reasoning in TQA can be enhanced by the self-consistency strategy. Moreover, the performance of LLMs also varies across different formats of the structured data. [117] transforms tables into eight formats, including but not limited to HTML, JSON, and Markdown. From the perspective of data, researchers designs analysis on the basis of self-supervised components in the structures [117, 120, 183]. For example, [183] conduct component-level perturbation (e.g, column adding), [120] investigate via seven structural understanding tasks (e.g., cell lookup and row retrieval) based on different components. Additionally, LLM capability evaluations consider structured knowledge from the scientific domains [37, 142] to the financial domain [48].

2.1.2 Graphs and Knowledge Graphs

Graphs and knowledge graphs (KGs) are also common formats for containing structured relationships among entities in diverse domains, ranging from recommender systems and scientific discovery. In a graph, entities (e.g, users in recommender systems) are modeled as nodes and their relationships (e.g., interactions with items) as edges, enabling structured reasoning and relational inference. Knowledge graph, as a special types of graphs, enriches graph structured with semantics information, each edge in knowledge graph typically represented as triples (head entity, relation, tail entity).

To support the development and evaluation of deep learning models on graph-structured data, a variety of benchmark datasets have been proposed, encompassing different types and domains of graphs. General-purpose knowledge graphs such as Freebase and DBpedia provide rich semantic structures over entities and relationships, enabling reasoning in open-domain settings. For more specialized applications, BioKG [180] offers a large-scale biomedical knowledge graph that integrates multiple curated databases to support complex biological and medical reasoning. Similar to foundational tasks like Tabular Question Answering (TQA) or Table-to-Text generation in structured tabular data, benchmarks for graph-based learning are typically organized around three primary tasks: node classification, link prediction, and graph classification. For example, Cora [89], Pubmed [114], and ogbn-arxiv [51] are widely used for node classification and link prediction, MUTAG [30], PROTEINS [11] are commonly used for graph classification.

Although the capabilities of machine learning and deep learning models on structured knowledge have been extensively evaluated through existing benchmarks and empirical studies, there remains a need for a unified testbed to assess whether more advanced models—such as large language models (LLMs)—can effectively memorize and reason over structured knowledge, including both tables and graphs, and achieve performance comparable to that on unstructured knowledge.

2.2 Representation Methods

To effectively reason over structured knowledge, it is crucial to represent the structured knowledge in the latent space of deep models. The representation level serves as the foundation for further reasoning in downstream applications, influencing the ability of a model to capture semantic relationships, structural dependencies, and entity interactions. In this section, we review the existing representation methodologies from challenges

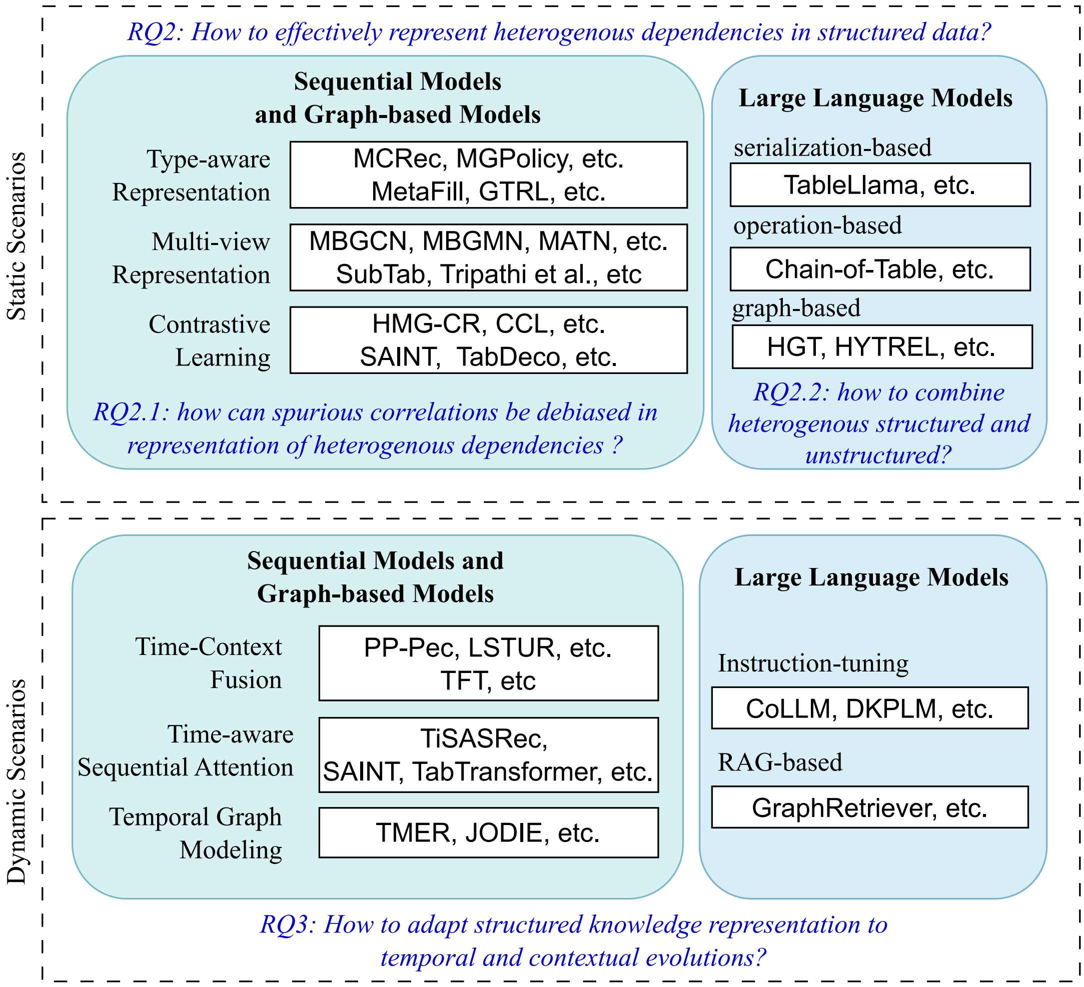


Figure 2.1: Overview of categories of existing structured knowledge representation methods.

in static scenarios (Section 2.2.1) to dynamic scenarios (Section 2.2.2) with examples of specific tasks and applications.

2.2.1 Static Heterogenous Dependencies

In static structured data, heterogeneous dependencies are common. For example, tabular data often contains hierarchical relationships, while interaction graphs include multiple types of edges. These diverse dependencies provide useful contextual signals but can also introduce noise, making representation and reasoning more challenging. In contrast to homogeneous structures, where one type of dependency is dominant, such as the sequential relations between tokens in natural language, heterogeneous settings require deep models to distinguish the semantics of different dependencies. We then examine current methods for handling heterogeneous dependencies in both conventional neural networks and LLMs, respectively.

2.2.1.1 Heterogenous Dependencies in Conventional Neural Networks

A key challenge in conventional neural networks (e.g., RNNs, Transformers, GNNs) lies in accurately representing heterogeneous dependencies. As illustrated in Fig. 2.1, methods for addressing heterogeneous dependencies can be broadly classified into three categories: type-aware representations [86, 96, 124, 145], multi-view representations [59, 132, 135, 159, 160], and contrastive learning [10, 20, 118, 164]. This section elaborates on these three categories of methods using detailed example models in recommendation systems and tabular data, as interaction graphs and tables are two common types of downstream tasks that involve structured knowledge.

Models that rely on Type-aware Representations use type encoding to enable distinction among different types of dependencies. Heterogeneous dependencies are often defined as various types of meta-paths within structured data. A typical example is the widely used Heterogeneous Interaction Networks (HIN). For instance, MCRec [96] predefines schemas for meta-paths in heterogeneous information networks and learns enhanced representations for users, items, and meta-paths, thereby improving recommendation accuracy. MetaFill [86] further advances this by redefining the meta-path identification problem to better exploit the semantics of HIN.

To handle data from different sources or relations, the second category of methods decompose the heterogeneous data into multiple subviews, each representing a specific type of dependency. These views are usually modeled separately and then fused for integration. Compared to the meta-path in Type-aware representaions, this method is able to explicitly represent the inter-dependencies relationships through calculating cross-attention. This category of methods is also widely used in recommendations and table-related reasoning. For example, MBGCN [59] uses separate Graph Convolutional Networks (GCN) to model different types of user-item interactions in recommendations; MATN [159] users a transformer-based encoder to capture cross-type behavior relations. In tabular tasks, SubTab [135] proposes a self-supervised framework to learn representations from different subset of features in each columns or rows of tables.

Contrastive learning has also proven effective in modeling the heterogeneity of structured data. Methods in this category construct contrasting representations to capture different types or strengths of dependencies, such as semantic versus structural relationships, or varying contextual information. For example, HMGCR [164] employs a contrastive learning objective to distinguish between different types of user behaviors in recommendation systems, while SAINT [118] adopts a novel contrastive self-supervised strategy to enhance its attention mechanism and better capture dependencies between rows and columns.

2.2.1.2 Heterogenous Dependencies in Large Language Models

Beyond conventional neural networks, large language models (LLMs) have demonstrated superior performance thanks to their massive parameter sizes and extensive pretraining on unstructured textual data. However, given that most of their training data is unstructured in nature, it raises the question of whether LLMs can achieve comparable performance on structured data such as tables and graphs. Representing heterogeneous dependencies inherent in structured knowledge poses a fundamental challenge for LLMs, as it contrasts with the sequential and often unstructured nature of user-inputted queries.

Prior efforts to enhance LLMs' capabilities in handling structured knowledge can be broadly categorized into three main approaches: serialization-based methods [43, 57, 91], operation-based methods [58, 68, 87, 148, 169], and graph-based methods [15, 19, 60].

Serialization-based methods convert structured data into a linear sequence of tokens, similar to how unstructured textual data is formatted for input into LLMs. TableLlama [178], a pioneering approach to enhancing LLMs' performance on tabular data, is fine-tuned on the proposed TableInstruct dataset, which comprises serialized tables and task-specific instructions for several representative tabular tasks. However, when dealing with highly complex tables or graphs, inquiry-relevant knowledge may be overlooked within the excessively long serialized token sequences [76, 181].

The second category of methods resort to one or a series of operations such as SQL queries to help LLMs reason over structured data [58, 68, 87, 148, 169]. For example, Chain-of-Table [148] iteratively samples operations to select specific portions of the table that are tailored to the inquiry. Dater [169] transforms the sub-questions generated by CodeX [17] into SQL queries, enabling step-by-step multi-hop reasoning. Although these operation-based methods effectively locate the inquiry-relevant knowledge from structured data, they struggle when the target cell or neighboring cells contain missing or incomplete information.

As messages propagate through the structures in Graph Neural Networks (GNNs), efforts have been made to integrate GNNs with LLMs to address structured knowledge more effectively [15, 53, 83, 110, 126]. For example, Chai et al. [15] uses a transformer module to encode the structured knowledge in graphs as the prefix of inputs to the LLMs. Additionally, graphs serve as powerful tools for representing tabular data [19, 60]. HGT [60] explicitly models tables as graphs by connecting various components within the tables to enhance LLM capabilities. Furthermore, HYTREL [19] is particularly relevant to our as it also empolys hypergraphs to represent tabular data, but it overlooks incorporating the semantics of task within prompts during message propagation. Existing works, while effective, primarily focus on utilizing LLMs rather than improving their inherent capabilities with model-agnostic modules. To the best of our knowledge, we are the first to leverage hypergraphs to enhance the capabilities of LLMs in handling structured knowledge.

2.2.2 Dynamic Evolution Adaption

The representation learning of structured knowledge are mostly focused on encoding entities and their relationships in static scenarios. However, in real-world applications, structured knowledge is inherently dynamic - evolving over time and shaped by varying contextual information such as user intent, location, and domain-specific semantics. Adapting deep models to these evolutions necessitates that researchers effectively take temporal and contextual dimensions into account in structured knowledge representations.

2.2.2.1 Structured Dynamics in Conventional Neural Networks

Conventional deep models based on RNN and transformer [136] often involves temporal and contextual evolution with a architectural changes that encodes these dynamics over time. As shown in Fig. 2.1, we categorize the conventional neural networks that consider the temporal and/or contextual evolution into three classes: Time-Context Fusion [5, 79, 106], Time-aware Sequential Attention [54, 72, 118], and Temporal Graph Modeling [16, 65, 112].

The first category of methods intuitively encodes additional dynamic factors (such as timestamps, locations) into the latent space as embeddings, which are then merged with structured inputs. News recommendation is a typical scenario where both temporal and new contextual shifts significantly influence the final predictions. For example, PP-Rec [106] encodes time-related factors (i.e., item popularity and recency) into latent space and fuses them with the user representation learned from structured user-item interaction data. In the case of tabular data, another common form of structured inputs, TFT [79] serves as representative model that combines static features with time-varying sales data, enabling interpretable modeling of dynamic feature modeling and achieving strong performance in tabular time-series prediction.

Time-aware attention mechanisms play an important role in adapting to temporal and contextual changes. Typically integrated into sequential models, these mechanisms focus on temporal sequences of interactions or records, using time-position encodings to enhance representation learning. TiSASRec [72] incorporates both time interval and positional information into self-attention layers, allowing the recommender to differentiate between short-term and long-term user preferences. In the context of time-indexed tabular data, temporal information is often treated as additional input features, with attention mechanisms computing dependencies between temporal dynamics and contextual features to facilitate integration. Time-aware transformers like SAINT [118] and TabTransformer [54] employ row-wise attention to capture dependencies along the time.

Given the strong performance of GNNs in modeling structural information, temporal graphs have been proposed to capture the evolution of structured knowledge, where nodes and edges can change over time. These methods typically decompose dynamic graphs into a sequence of time-ordered subgraphs. In the recommendation domain, JODIE [65] models the co-evolution of users and items by updating their representations through recurrent networks triggered by user-item interactions. DySAT [112] further integrates structural and temporal attention mechanisms to jointly capture both topological and temporal patterns. TMER [16] leverages dynamic knowledge graphs with attention mechanisms to sequentially model the evolving relationships between users and items, while also enhancing explainability through temporal information.

2.2.2.2 Structured Dynamics in Large Language Models

To explore the limits of Large Language Models (LLMs) in handling structured knowledge, it is also important to consider how well they perform in modeling structured dynamics, which refers to the evolution of structured knowledge across time or context. Research in this area is still at an early stage. One of the most direct approaches to incorporating structured knowledge into LLMs is to inject structural representations into prompts, which resembles instruction tuning. For example, CoLLM [179] capture collaborative signals in recommendations by first modeling the structured interaction data with traditional collaborative-aware models such as LightGCN [42], and then integrate the resulting embeddings into the latent layers of LLMs through prompt-based inputs. Another direction of research focuses on simulating dynamic structural evolution using retriever augmented generation (RAG) methods. For example, GraphRetriever [92] elects subgraphs from the full knowledge graph and constructs task-specific structured inputs dynamically. This enables LLMs to better adapt to varying structural dependencies across different reasoning tasks.

Chapter 3

Structured Knowledge Benchmark

At the data-level, structured knowledge pertain unique characteristics differentiates from unstructured data such as texts which are usually used to training deep models, especially language models. The following sections describe how StructFact was developed based on its unique characteristics, along with our in-depth analyses and experimental insights, with a focus on factual knowledge in structured data.

3.1 Unique Charateristics of Structured Knowledge

Compared to unstructured data, certain unique characteristics of structured data affect the ability of LLMs to understand and reason about factual knowledge [33]. These characteristics include: (1) *Heterogeneity*. structured knowledge consists of diverse data types (e.g., texts, numerics, dates). Misunderstandings or biases of any type can lead to inaccuracies in the factual knowledge. (2) *Topological Interdependencies*. Most LLMs are based on the Transformer architecture [136] and are trained with a next-word prediction loss objective, primarily designed to process continuous text data. Extracting relevant interdependencies from complex topological structures is a significant challenge for LLMs in understanding and reasoning about facts. (3) *Order Invariance*. A key assumption in pretraining is that the order of words significantly impacts their semantics [19]. However, in structured data, the permutation of entities (e.g., rows or columns in a table) does not alter the underlying factual knowledge. (4) *Sparsity*. To maintain the same performance in sparse structured data (e.g., missing values or incomplete descriptions) as in data-rich scenarios, LLMs need to accurately utilize the general knowledge learned during pretraining and avoid non-factual imputations. (5) *Lack of Prior Knowledge*. Structured data holds domain-specific knowledge not exposed during pretraining, challenging the accurate application of general reasoning to downstream tasks without distortion [27, 69, 182]. These characteristics of structured data impact the ability of LLMs to reason about factual issues, limiting their real-world applications, especially in high-risk domains such as healthcare and finance. To enable LLMs to effectively utilize knowledge embedded in structured data and enhance reliable reasoning, it is essential to examine their capabilities based on the specific characteristics of structured data.

In light of these characteristics, we specifically focus on the structural challenge posed in previous works represented by the Pinocchio benchmark [52], and analyze the reasoning capabilities of LLMs on structured data from the perspective of five factual tasks: Arithmetic Calculation, Geography-Time Reasoning, Multi-hop Reasoning, Composition Understanding, Combining Structured and Unstructured. We develop StructFact, a benchmark comprising 13,407 factual queries and corresponding evidence in various structures (i.e., tables, lists, and graphs), covering diverse data types, knowledge domains, timeliness, and regions. We categorized these questions into five factual tasks and provided fine-grained difficulty annotations based on the specific focus of each task to facilitate a multifaceted analysis. Additionally, to specifically test the capability of LLMs to reason over fresh structured facts not encountered during pretraining in real-world applications, we have developed StructFact-Unseen.

Through explorations with StructFact, we examine how 10 commonly used LLMs understand and reason with factual knowledge stored in structured data. For instance, in tasks that rely on *heterogeneous* data, LLMs heavily depend on the order of information for understanding. We aim for StructFact to serve as a compass in exploring the boundaries of LLMs in knowledge-sensitive tasks involving structural facts, while also advancing their practical applications in real-world scenarios.

3.2 Dataset Construction

To assess LLMs' reasoning abilities over structured facts, we define five tasks based on characteristics of structured data and carefully curate factual questions supported by diverse data types.

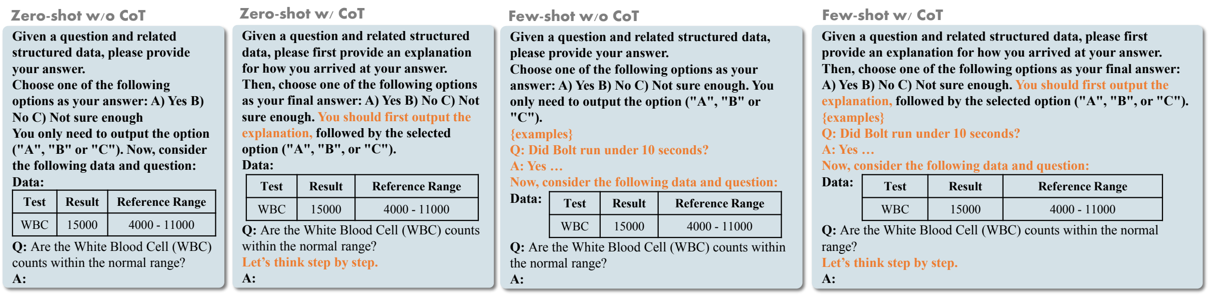


Figure 3.1: Prompts used in different settings (main differences with zero-shot w/o CoT are marked in orange).

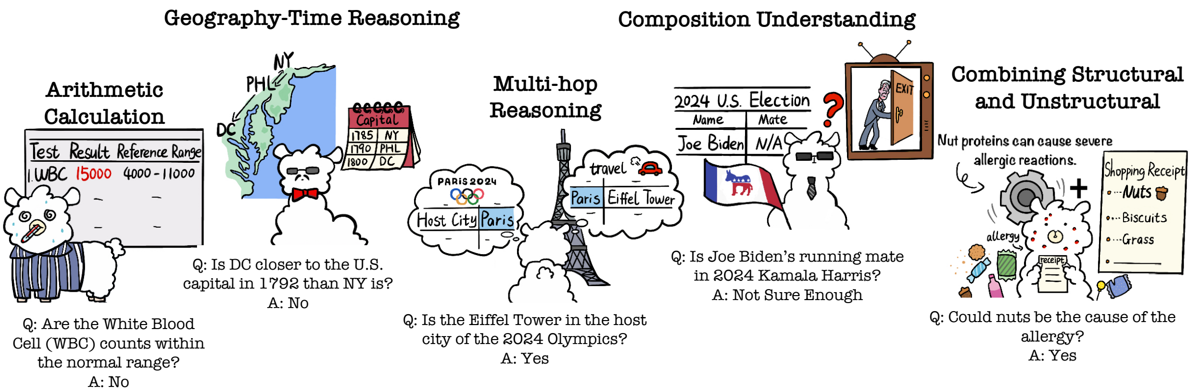


Figure 3.2: StructFact evaluates LLMs' reasoning ability over structured factual knowledge across five tasks.

3.2.1 Tasks

We design five tasks aimed at investigating different research questions to thoroughly assess the ability of LLMs on structural data.

Arithmetic Calculation. Given the substantial amount of numerical facts stored in structural data, such as the health report in Figure 3.2, LLMs are required to perform arithmetic calculations over such *heterogeneity* (i.e., text and numeric) [4, 55]. Models such as Graph Neural Networks (GNNs) seamlessly handle arithmetic calculations by inferring arithmetic rules from numerical patterns through their structural architecture, while LLMs are based on the transformer architecture which is designed for unstructured data. Moreover, structured data containing numbers (e.g., clinical sheets) is usually confidential and *lacks prior inclusion* in the LLMs' knowledge base.To evaluate the capability of handling the *lack of piror knowledge* and *heterogenity*, especially numerical types, StructFact includes factual questions from simple numerical matching to difficult computational analysis.

Geography-Time Reasoning. Geographic and temporal information often appears simultaneously in structured data and can be presented in various formats. Geographical data encompass geographic coordinates (such as latitude and longitude), city names, and country names, whereas temporal data include dates and time periods. This *heterogeneity* challenges LLMs to precisely interpret and align these diverse formats. For example, to determine whether Washington D.C. or New York was closer to the U.S. capital in 1792, as shown in Figure 3.2, LLMs must combine the year information of the U.S. capital with the geographic coordinates of these cities. To evaluate the proficiency of LLMs in geography-time reasoning, we integrate factual knowledge pertaining to geographical, temporal, and geography-time entities.

Multi-hop Reasoning. Factual knowledge in structural data involves entities dispersed across multiple sources [75]. In Figure 3.2, the query from the tourist llama involves structured knowledge about the Olympics and travel guides. However, language models typically generate answers by gathering factual knowledge separately, thereby overlooking the *topological interdependencies* [166]. When gathering data from multiple sources, models should recognize the *order invariance* of structural data. Unlike textual data, which is order-dependent, the order of entities within a similar topological structure should not affect the inherent factual knowledge. In StructFact, we include questions where knowledge is spread across multiple discontinuous structured sources.

Composition Understanding. Reasoning about factual knowledge in structural data suffers from the *sparsity* issue due to missing values or incomplete descriptions. Beyond that, LLMs are expected to accurately reason through these sparse information while without misinterpreting *topological interdependencies*. As Figure 3.2 displays, to answer the question “Is Joe Biden’s running mate in 2024 Kamala Harris?”, LLMs have to comprehend the header “2024 U.S. Election”, which spans multiple columns with a missing value denoted by “N/A”. To assess whether LLMs comprehend structural compositions, StructFact includes factual questions about components with missing data, complex structures, and incomplete descriptions.

Combining Structured and Unstructured. Given the *sparsity* and *lack of prior knowledge* of the domain-specific information in structured data, LLMs needs to fully leverage the factual knowledge learned from textual contexts. The knowledge presented in unstructured data (e.g., table captions) often provides an important context for understanding the knowledge in structured data. Moreover, the general knowledge base of LLMs aids in reasoning domain-specific knowledge within structured data. As exemplified in Figure 3.2, general knowledge of nut proteins helps infer the cause of the allergy from the shopping receipt. To assess how structured data combined with their unstructured contexts in LLMs, StructFact includes factual questions that require factuality verification spanning both structured and unstructured evidences.

Models	Zero-shot w/o CoT		Zero-shot w/ CoT		Few-shot w/o CoT		Few-shot w/ CoT		Overall	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Qwen2-7B	29.94	37.73	46.77	50.61	44.69	49.51	52.74	56.63	43.53	48.62
LLaMA-3-8B	28.39	33.32	26.66	35.03	26.43	33.22	49.02	50.29	32.63	37.97
Gemma-2-9B	22.83	26.36	43.72	49.58	34.89	40.19	55.97	56.19	39.35	43.08
Qwen2-7B Instruct	46.67	52.40	43.22	51.47	44.18	50.24	43.40	51.95	44.37	51.52
LLaMA-3-8B Instruct	60.62	58.23	43.37	50.19	61.07	59.15	49.54	57.80	53.65	56.34
Gemma-2-9B It	42.66	49.64	47.63	56.97	45.92	53.50	49.78	58.87	46.50	54.75
GLM-4-9B Chat	56.23	59.04	48.44	57.36	54.91	58.15	52.37	60.11	52.99	58.67
Mistral-7B Instruct	47.63	50.45	40.52	49.75	55.78	57.41	46.96	55.49	47.72	53.28
GPT-3.5-turbo	59.84	60.76	59.90	63.94	52.30	56.84	63.91	66.52	59.00	62.02
GPT-4o-mini	**65.12**	**67.08**	**62.96**	**68.11**	**61.44**	**65.03**	**64.96**	**69.58**	**63.62**	**67.45**

Table 3.1: Performance of 10 LLMs on the benchmark using various prompts.

Tasks	Distribution			
	Fact.	**Non-Fact.**	**NEI**	**Overall**
Arithmetic Calc.	1,438	1,235	69	2,742
Geography-Time Reas.	1,602	1,717	112	3,431
Multi-hop Reas.	1,694	1,621	79	3,394
Composition Und.	683	925	24	1,632
Struct. and Unstruct.	1,226	913	69	2,208
Total	**6,643**	**5,411**	**353**	**13,407**

Table 3.2: The statistics of our benchmark.

3.3 Main Results

To examine the factual reasoning capabilities of LLMs on structured data, we conduct experiments with StructFact across 10 LLMs trained through pretraining, instruction tuning, and reinforcement learning with human feedback (RLHF). Given the bias of LLMs towards "Yes" answer [104, 184, 185], we replaced the Fact./Non-Fact./NEI options with A/B/C, respectively[1] and report each model's average performance with respect to weighted accuracy and F1 score over three runs with varied option orders.

[1]For pretrained LLMs, due to their weak instruction-following capabilities, outputs beyond 'A/B/C' are considered misclassifications, such as a response of 'None', and categorized as False Negatives.

Models	Arithmetic Calc.		Geography-Time Reas.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Qwen2-7B	27.12	34.60	28.22	34.77	29.31	38.66	31.68	39.51	35.78	43.17
LLaMA-3-8B	27.77	32.25	28.12	31.99	28.07	33.98	28.51	33.91	30.00	34.95
Gemma-2-9B	17.01	23.47	21.30	24.00	25.92	27.56	26.39	27.83	25.10	28.82
Qwen2-7B Instruct	52.41	54.01	41.41	47.41	41.96	49.87	50.67	56.25	52.02	57.76
LLaMA-3-8B Instruct	53.63	46.20	54.66	50.51	**68.00**	68.89	65.70	66.12	63.46	58.73
Gemma-2-9B It	49.37	51.40	34.89	40.88	37.69	47.74	49.28	56.04	49.16	57.38
GLM-4-9B Chat	54.59	52.69	47.74	51.63	60.37	64.53	65.05	65.74	58.56	61.99
Mistral-7B Instruct	52.64	51.87	43.77	46.30	43.36	47.45	46.94	49.79	54.48	57.81
GPT-3.5-turbo	57.70	57.90	57.42	58.82	62.82	63.26	63.79	64.09	58.79	60.28
GPT-4o-mini	**63.93**	**64.15**	**59.92**	**62.96**	67.56	**69.66**	**71.88**	**72.44**	**65.94**	**68.50**
Overall	45.62	46.85	41.75	44.93	46.51	51.16	50.00	53.17	49.33	52.94

Table 3.3: Performance of 10 LLMs on the across five tasks under the zero-shot setting without CoT.

3.3.1 Different Prompts

In Table 6.2, our prompts adhere to the input formats used in previous studies [117, 120, 147], where factual questions from StructFact are combined with corresponding structured data and fed into these LLMs, prompting the models to answer the questions. From the results reported in Table 6.2, we conclude the following findings.

- From an overall standpoint, **the ability to comprehend instructions, facilitated by either a larger parameter size or instruction tuning, contributes more significantly to a model's reasoning over structured data than pretraining knowledge alone.** GPT-3.5-turbo, despite its outdated knowledge cutoff in 2021, achieves competitive performance thanks to its large parameter size. Also, models with instruction tuning exhibit superior results compared to the pretrained models. The results obtained by LLaMA-3-8B Instruct, Gemma-2-9B, and Qwen2-7B Instruct outperform their corresponding pretrained models, with an average accuracy improvement of 22.93%. Those pretrained LLMs perform even below the level of random guessing (33.33%) in the zero-shot setting without CoT, as they struggle to follow instructions.

- **For pre-trained models, both the Chain of Thought (CoT; [150]) and few-shot strategies helps in utilizing their factual knowledge.** In a zero-shot setting without CoT, the performance of pre-trained models falls below random guessing (with a probability of 33%); incorporating few-shot learning and CoT results in an average F1 score improvement of 5.53%. **The CoT strategy has even negative impact on some instruction-tuned models (e.g., LLaMA-3-8B Instruct), and few-shot examples yield limited improvements.** More complex prompting strategies also result in modest gains in instruction-tuned models.

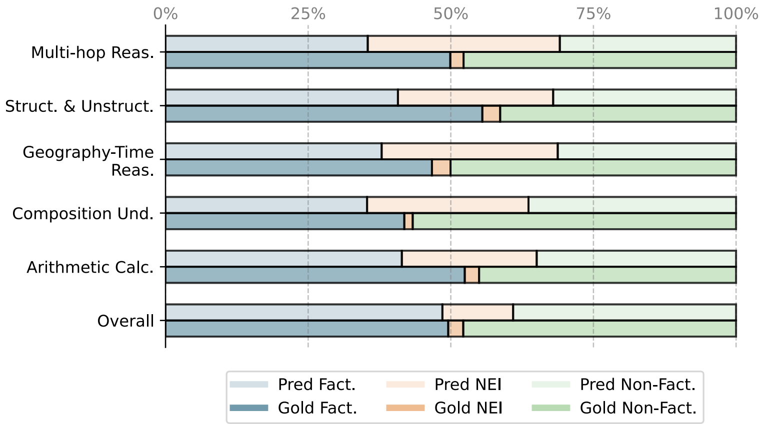


Figure 3.3: The distribution of three types of responses across five tasks, averaged across 10 LLMs.

3.3.2 Different Tasks

We further analyze the reasoning performance of the selected 10 LLMs on structured data from the standpoint of the five tasks. From the results in Table 3.3, we have the following observations.

- LLMs perform inferior on the tasks of geography-time reasoning and arithmetic calculation, with average weighted F1 scores of 44.93% and 46.85%, respectively. **This underperformance can be attributed to the limitations of LLMs in processing *heterogeneous* evidence**, i.e., reasoning collectively over dates, numbers, and/or texts within structured data sources.

- Among the five factual tasks, LLMs perform relatively well on the tasks of Composition Understanding and Combining Structured and Unstructured data. After a detailed examination of the cases, we conclude that this benefit stems from **LLMs utilizing their in-context learning and extensive commonsense knowledge to overcome the *sparsity* issue in the structured evidence for these two tasks**.

- We further analyze the distribution of three labels in Figure 3.3 and observed that the proportions of predicted NEI labels are generally higher than those of the gold labels across the five tasks, the proportions of factual and non-factual responses vary between different tasks. This indicates that, **akin to human behaviors [134], LLMs demonstrate caution when accepting or rejecting factual queries, when comparing to the ambiguous answer.**

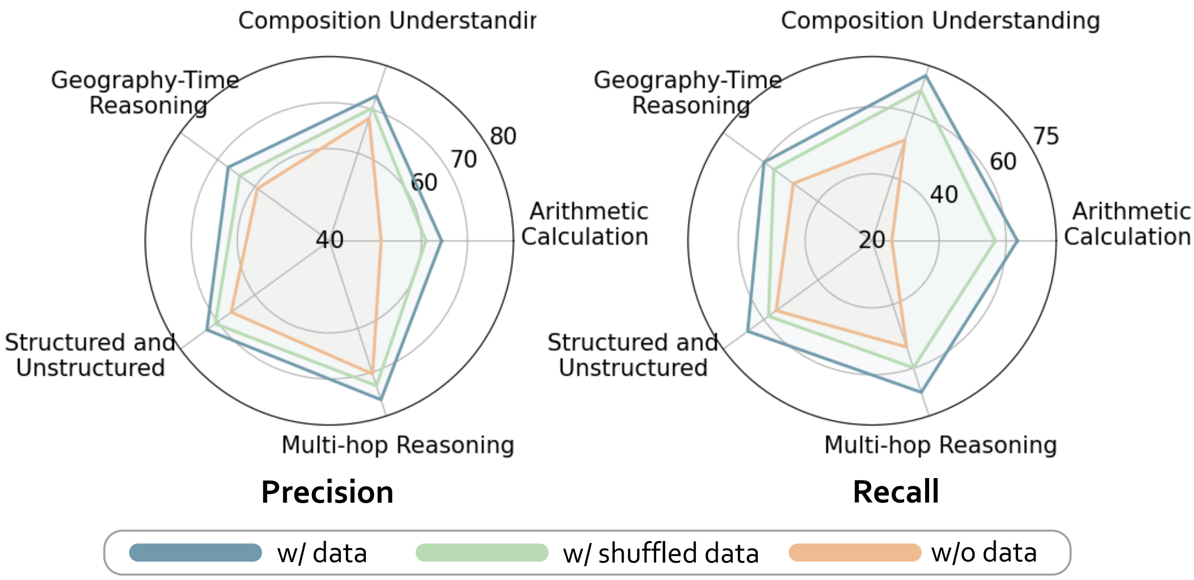


Figure 3.4: Performance of GPT-4o-mini under different settings of structured evidence.

3.4 Analysis

Building on the main results, we perform in-depth analyses from various perspectives to assess the LLMs' abilities in completing the five tasks, with GPT-4o-mini (which achieves best in experiments) as the representative model. Analyses of other LLMs are included in Appendix A.11.

3.4.1 Resilience to Evidence

We first investigate whether the ability of LLMs to answer factual questions is influenced by the presence of structured evidence. To this end, we categorize the resilience of LLMs to evidence into three levels, ranging from stringent to adaptable: (i) efficiently understanding and reasoning with the provided structured data as evidence, (ii) adapting to irrelevant interventions in the structure of the evidence data, and (iii) accurately recalling prior general knowledge without the support of structured data. We expect LLMs to maintain strong performance across all three levels, indicating remarkable resilience.

To this end, we assess the performance of GPT-4o-mini under three distinct conditions: (i) with structured data provided as corresponding evidence for the factual questions (denoted as "w/ data" in Figure 3.4), (ii) with the structured data shuffled (denoted as "w/ shuffled data" in Figure 3.4), and (iii) without any structured data as evidence (denoted

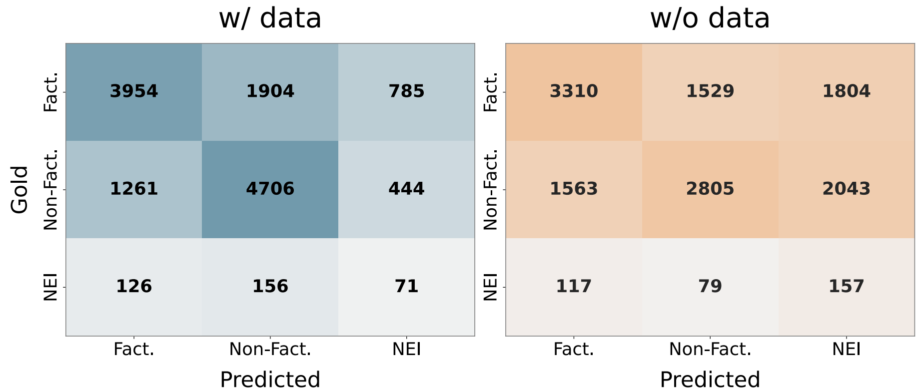


Figure 3.5: Confusion matrices of performance under the settings w/ and w/o structured data as evidence.

as “w/o data” in Figure 3.4). The first condition aligns with the zero-shot without the CoT setting in the main results (Table 6.2). In the second condition, we exploit the *order invariance* property of structured data to introduce semantically irrelevant interventions by shuffling the rows and columns in tables and the elements in lists in our StructFact benchmark. For the third condition, since all factual questions in our framework are supported by structured data from Wikipedia, we anticipate that the LLM will rely on its pretraining knowledge to effectively handle scenarios where evidence is absent. We assess the resilience to evidence of the LLM across the five tasks under these three conditions, and depict in Figure 3.4 and Table A.18.

Transitioning from the original structured data (w/ data) to shuffled structured data (w/ shuffled data) results in only a marginal performance decrease, with the largest decline of 3.2% in precision in the Multi-hop Reasoning task, where *order invariance* matters. **This demonstrates the strong adaptability of LLMs to the *order invariance* characteristic of structured data**, and also validates our task categorization.

Transitioning from evidence-rich scenarios (w/ data) to the evidence-absence scenario (w/o data), the LLM’s performance drops significantly, especially in the tasks of Arithmetic Calculation and Combining Structured and Unstructured, which underscores the *lack of prior knowledge* characteristic. We further analyze this performance drop with the confusion matrices in Figure 3.5 and attribute it to the uncertainty and a higher tendency for the model to classify facts or non-facts as NEI. This shows that **LLMs do not possess adequate *prior knowledge of facts* contained in real-world structured data.**

3.4.2 Fine-grained Studies of Different Tasks

In this section, we fine-grainedly assess GPT-4o-mini's reasoning on structures across five tasks. See Appendix A.7 for the fine-grained categorization defined for each task.

3.4.2.1 Arithmetic Calculation

To assess whether large language models (LLMs) are capable of capturing and memorizing arithmetic rules, we categorize the questions in the arithmetic calculation task into three levels of mathematical problems depending on varying degrees of arithmetic difficulty: numerical matching, numerical comparison, and computational analysis. For instance, the factual question "Are the White Blood Cell (WBC) counts within the normal range?" as illustrated in Figure 3.2, falls under the numerical comparison category. Figure 3.6(a) presents the performance of GPT-4o-mini across these three categories of mathematical problems. This suggests that **LLMs manage basic numerical tasks like matching and comparison effectively, but struggle with complex computational analyses, such as statistics analysis.**

3.4.2.2 Geography-Time Reasoning

As shown in Table 3.3, LLMs exhibit inadqequate performance in the Geography-Time Reasoning task. We conducted a detailed analysis of GPT-4o-mini's performance across different named entity categories. In Figure 3.6(e), we classified the Geography-Time Reasoning questions in StructFact into three categories: (i) temporal, which includes questions about dates (DATE), and times (TIME); (ii) geographical, encompassing questions related to political regions such as countries and cities (GPE), as well as locations such as mountains and rivers (LOC), and artificial landmarks (FAC); and (iii) geography-time, which involves questions containing both geographical and temporal entities (DATE+GPE, DATE+LOC, DATE+FAC). Overall, **the LLM performs consistently at understanding and reasoning with geography-time knowledge than with data that only involves temporal or geographical entities**. The varying performance across different entity types suggests that the LLM is more effective with entities that offer detailed granularity in geographical dimensions.

3.4.2.3 Multi-hop Reasoning

To investigate the capability of LLMs in recognizing and combining knowledge from various discontinuous sources of structured data, we categorized factual questions in

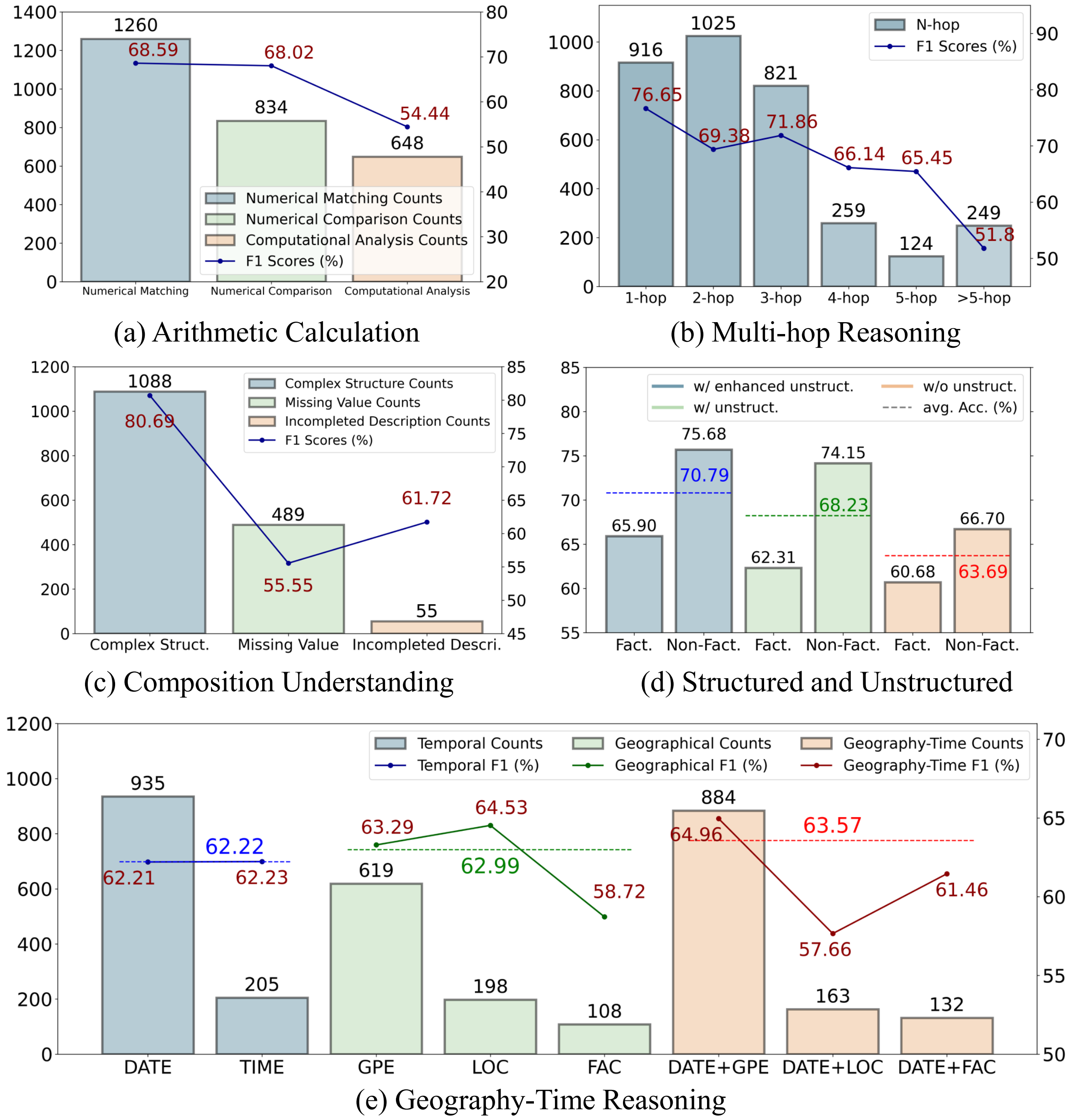


Figure 3.6: Fine-grained analysis of performance over the five tasks under zero-shot w/o CoT setting.

the Multi-hop Reasoning task at a more fine-grained level based on the number of hops required to arrive at an answer. A “hop” refers to the step in which the LLM needs to infer knowledge by combining knowledge from two data sources. In particular, in our analysis, each source is defined as a Wikipedia element (e.g., cells, headers, captions in tables, or items in lists) that serves as evidence supporting the gold answer. Figure 3.6(b) reveals a clear trend: **as reasoning tasks become more complex, requiring an increasing**

number of hops, the LLMs' effectiveness in reasoning over factual knowledge from structured data diminishes. Notably, there is a significant performance decline after 5-hop questions, with a 13.65% decrease in F1 score observed in questions with more than 5 hops.

3.4.2.4 Composition Understanding

To answer whether LLMs can accurately reason factual knowledge from challenging compositions in structured data, we categorize these compositions into three types of irregularities: (i) complex structure, where compositions involve intricate dependencies such as a single table cell spanning multiple columns; (ii) missing values, where cells contain unknown values; and (iii) incomplete descriptions, where cells have ambiguous or insufficient descriptions. Figure 3.6(c) illustrates illustrates that the primary bottleneck in enhancing LLMs' performance in understanding special composition within structures lies in addressing the challenges of missing values and incomplete descriptions. This challenge is associated with the characteristics of *lack of prior*, indicating that **accurately aligning general knowledge in LLMs with the domain-specific knowledge in structured data remains a significant obstacle for LLMs.**

3.4.2.5 Combine Structured and Unstructured

A prominent strength of LLMs in factual reasoning is their ability to comprehend knowledge in textual data. When extending this capability to tasks that involve structured data, it becomes imperative to assess whether LLMs can effectively combine factual knowledge extracted from unstructured contexts with reasoning applied to structured data. Therefore, beyond the original unstructured context provided as evidence in the Combining Structured and Unstructured task, we assess the capability of LLMs in scenarios with enhanced unstructured context, as well as in situations where unstructured context is absent. The results shown in Figure 3.6(d) illustrate that the performance of LLMs can be slightly improved by the availability of enhanced contexts when handling factual reasoning over structured data. It is noteworthy that in non-factual tasks, LLMs performed slightly better when provided with the enhanced unstructured context, compared to the original ones. The substantial decrease in performance when unstructured context is absent suggests that **LLMs are particularly dependent on this unstructured context for this task, especially in non-factual circumstances.**

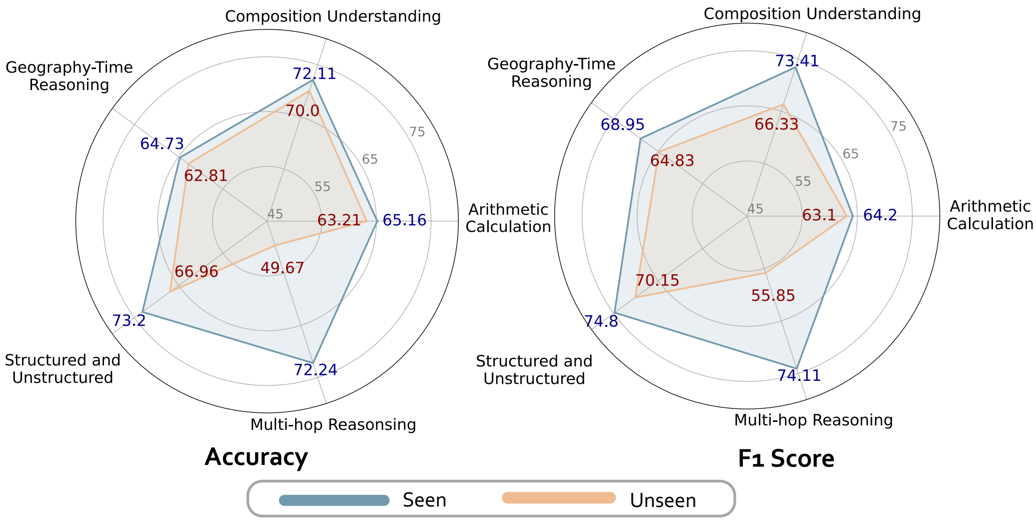


Figure 3.7: Performance of GPT-4o-mini with structured data seen and unseen during pretraining.

3.4.3 StructFact-Unseen

To assess the performance of large language models (LLMs) on data not encountered during their pretraining, we constructed StructFact-Unseen to periodically crawl the most recent structured data from new Wikipedia pages. The current StructFact-Unseen dataset, used in this submission, covers the period from December 15, 2024, to January 14, 2025. Please see Table A.3 for the pretraining knowledge cutoff dates of each selected LLM.

From Figure 3.7, it is clear that the language model (LLM) demonstrates a modest reduction in performance when processing structured data not previously encountered during its pretraining phase, compared to data it has been exposed to before. The LLM particularly maintains generalizable performance in tasks challenged by *heterogeneous* data, such as arithmetic calculation and geography-time reasoning. This generalization ability may be attributed to the domain-specific factual knowledge required for these tasks, which cannot be readily deduced using the general knowledge acquired from previously encountered data. Notably, there is a significant decline in performance on the StructFact-Unseen dataset for multi-hop reasoning. This suggests that **LLMs may rely more on their intrinsic knowledge base, rather than the structured evidence provided, to excel in multi-hop reasoning.**

Table 3.4: Performance of GPT-4o-mini using different prompting strategies across five factual tasks, with performance differences relative to Zero-shot w/o CoT.

Methods	**Arithmetic Calc.**		**Spatiotemporal Cogn.**		**Multi-hop Reas.**		**Composition Und.**		**Struct. & Unstruct.**		**Overall**	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Zero-shot w/o CoT	62.52 ↑0.0	63.23 ↑0.0	60.13 ↑0.0	63.33 ↑0.0	58.04 ↑0.0	63.72 ↑0.0	67.42 ↑0.0	67.96 ↑0.0	62.10 ↑0.0	66.88 ↑0.0	60.80 ↑0.0	63.67 ↑0.0
Self-Refine.	58.03 ↓4.5	60.73 ↓2.5	54.44 ↓5.7	60.08 ↓3.2	51.61 ↓6.4	58.81 ↓4.9	66.91 ↓0.5	68.17 ↑0.2	56.51 ↓5.6	63.37 ↓3.5	55.36 ↓5.4	60.38 ↓3.3
Self-Consis.	62.80 ↑0.3	63.49 ↑0.3	59.84 ↓0.3	63.02 ↓0.3	58.30 ↑0.3	63.92 ↑0.2	66.91 ↓0.5	67.35 ↓0.6	62.30 ↑0.2	67.06 ↑0.2	60.83 ↑0.0	63.69 ↑0.0
Format Instruct.	62.82 ↑0.3	63.44 ↑0.2	58.53 ↓1.6	61.95 ↓1.3	57.52 ↓0.5	63.10 ↓0.6	66.67 ↓0.7	67.16 ↓0.8	61.37 ↓0.7	66.38 ↓0.5	60.03 ↓0.8	63.03 ↓0.7

3.4.4 Analysis towards Other Prompting Strategies

Given the successes of other CoT strategies and input data format instructions [**?**], we are interested in exploring their impact on reasoning about factual knowledge within structured data. We include three prompting strategies: (i) self-refinement [88], which guides the LLM to iteratively evaluate and refine its previous responses to reach the correct answer, (ii) self-consistency [146], which mitigates hallucination through majority voting on multiple responses from the LLM, and (iii) format instructions, which prompts with descriptions of the format of the inputted structured data. There are the following notable observations from the results in Table 3.4. i) Self-consistency marginally improves performance across five tasks, with an overall enhancement of 0.23%, compared to the zero-shot results without CoT. ii) Format descriptions help the LLM better interpret numerical compositions, leading to a 1.02% improvement in accuracy on Arithmetic Calculation tasks.

3.5 Limitations

In this work, we present StructFact, a benchmark specifically developed to assess the factual reasoning abilities of LLMs over structured data. StructFact comprises 13,407 questions spanning five factual tasks built upon the characteristics of structured data. We tested 10 LLMs with StructFact and observed that they struggle with reasoning over *heterogeneous* structured data, especially in complex arithmetic operations.

While this study offers a valuable benchmark for evaluating LLMs' reasoning capabilities with structured data, it presents this data in a uniform format—markdown for tables and triplets for graphs. The omission of other structured formats such as JSON and XML could limit the generalization and applicability of . Second, the uneven distribution of fine-grained labels in each task could affect the analysis. These limitations potentially prevent this work from broad applications and should be further improved to allow real-world deployment scenarios.

Chapter 4

Debiasing Heterogeneous Dependencies

Knowledge in structured data is organized through various relationships. These dependencies offer valuable context for representation learning in deep models. However, despite being carefully curated by experts, spurious dependencies may still be present, potentially leading to biased representations. This chapter introduces a causal-enhanced model to debias the confounding effects result from the spurious dependencies in users' multi-typed behaviors.

4.1 Challenge in Multi-behavior Dependencies

Recommendation systems have become a core in a wide range of online platforms by aiding users' decision-making in overwhelming resources provided. Multi-behavior recommendations (MBRs) go beyond singular-typed interactions, as they aim to enhance recommendation performance by capturing the intricate semantics from multi-typed interactions [59, 160]. From a causal perspective, the predictions of MBR can be viewed as the outcomes, given varied users' behavioral data serving as the treatment.

MBR systems improve user modeling but are more vulnerable to including spurious correlations caused by unobserved confounding factors. As demonstrated in Figure 6.1 (a), user moods that impact both user behaviors (treatment) and model predictions (outcome) are in fact confounders in MBR. Particularly, when users are stressed, they are more likely to make impulse purchases, leading to fewer *favorite* or *add to cart* behaviors (treatment) and increase the probability of buying (outcome). The presence of

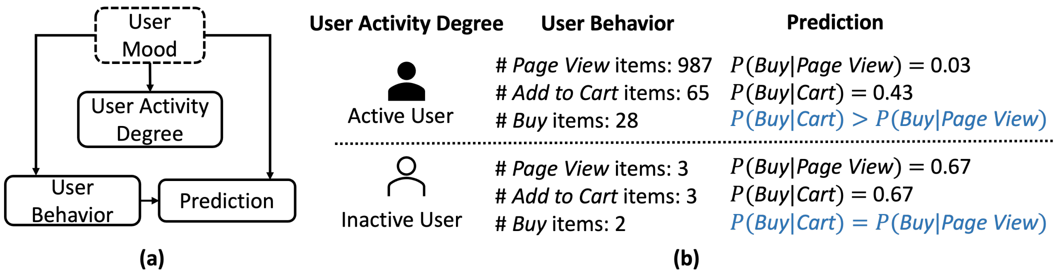


Figure 4.1: (a) Causal model of user mood (unobserved confounder), user behavior (treatment), prediction (outcome), and user activity degree learned from interactions. (b) A toy example of stratifying users by their activity degrees MBR: the left column refers to stratification of users by their activity degrees, the middle column displays the number of items being interacted with, while the right column is the probability of *buying* conditioned on *page view* and *add to cart*.

unobserved confounders can introduce spurious correlations, negatively impacting the performance of MBR models. For instance, a skewed model might incorrectly assume that minority behaviors like "*favorite*" and "*add to cart*" don't genuinely represent a user's intent to purchase, which goes against our understanding. In practice, these unobserved confounders are unmeasurable due to challenges of technical complexities and privacy considerations [177, 190].

Causal reasoning sheds light on this challenge, where the frontdoor adjustment [161, 190] and proxy variables [177] emerge as two dominant approaches to mitigate the negative impacts of unobserved confounders. Nevertheless, these two methods pose practical challenges in real-world recommendation systems due to their stringent assumptions and restricted data. On one hand, the validity of the front-door adjustment relies on strong assumptions regarding an intermediate mediator between the treatment and recommendation outcomes. Identifying a mediator that satisfies such requisite assumptions presents a formidable challenge in MBR scenarios. On another hand, deriving proxy variables from additional information, such as user features, is not always available due to privacy considerations. In short, both the frontdoor adjustment and proxy variables relying on additional context information are less practical to be employed in real-world MBRs.

To mitigate these challenges, we proposed a Counterfactual-enhanced Multi-Behavior Recommender (C-MBR), which recommends items via stable counterfactual reasoning on multi-behavior interactions. Firstly, we explore the multi-behavior interaction data to define an observable proxy variable, which is correlated with unobserved confounder. Rather than depending on extra context information, a stable counterfactual reasoning method is designed to debias an unobserved confounders via the stratification of the proxy variable [138]. In particular, we take Figure 6.1 as an example, in which users' activity degree is captured as a proxy variable for the unmeasurable users' mood (i.e., unobserved confounder). This proxying is mainly because users may be more likely to actively engage in different interactions when they are in a cheerful mood. By proxying users' mood with activity degree, we are able to further analyze the confounding effects by dividing users into active and inactive stratum. As highlighted in blue in Figure 6.1 (b), active users tend to exhibit a higher probability to buy under the *click* behavior compared to that under the *add to cart* behavior, whereas that under these two behaviors is relatively even among inactive users.

We model the trend highlighted in blue in Figure 6.1 (b) with an ordering. In the scenarios of MBR, this ordering also indicates the buying inclinations of different user behaviors. Specifically, to improve user modeling, we utilize complex dependencies at both intra- and inter-behavior levels, and further enhance it through stable counterfactual reasoning while adhering to the ordering constraints.

The main contributions are summarized as follows:

- C-MBR is designed to debias the unobserved confounding effect on MBRs through stable counterfactual reasoning by stratifying a proxy variable.
- C-MBR explores the complex relationships among multiple behaviors from a 'what-if' perspective by conducting stable interventions based on intra- and inter-learning.
- We conduct extensive experiments on two real-world datasets and prove the effectiveness of stable counterfactual reasoning in debiasing unobserved confounders.

4.2 Problem Formulation

In this section, we first propose a stable counterfactual ordering for debiasing the negative effects of unobserved confounders. Then, we formulate the problem of conducting counterfactual reasoning for MBRs.

4.2.1 Stable Counterfactual Ordering

In MBR, the system makes predictions (**Y**) based on users' multi-behavior interactions (**B**), which are driven by user preferences (**U**). This process can be represented by

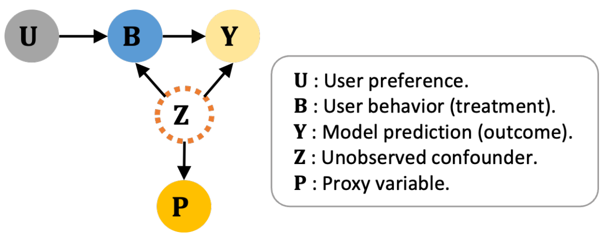


Figure 4.2: Causal model of multi-behavior recommendation. Solid circles and dashed circles represent observed and unobserved variables, respectively.

the causal path $\mathbf{U} \to \mathbf{B} \to \mathbf{Y}$ shown in Figure 6.2. We define the behavior set as $\mathcal{B} = \{b_1, ..., b_{|\mathcal{B}|}\}$ to include all the candidate values for $\mathbf{B}$ in the factual space, and $\{1, 0\}$ for $\mathbf{Y}$, where 1 represents the item v is included in the top-K recommendation list for user u, denoted by $v \in R_{u,K}$, 0 otherwise. For each behavior $b \in \mathcal{B}$, its counterfactual b^* is constructed by changing b at a minimal extent (e.g., fewer interactions under b), helping to answer the "*what-if*" question that *what happens to the recommender prediction* $\mathbf{Y}$ *if the behavior* b *changes to its counterfactual* b^*.

The unobserved confounder $\mathbf{Z}$ simultaneously influence the treatment ($\mathbf{B}$) and the outcome ($\mathbf{Y}$). A direct approach to debaise the unobserved confounder $\mathbf{Z}$ is stratifying users based on values of $\mathbf{Z}$. However, since $\mathbf{Z}$ is unobserved due to technical complexity and privacy concerns, we stratify a proxy variable $\mathbf{P}$ learned from historical multi-behavior interactions without any additional context information (e.g. user profile). For instance, users can be divided into two stratums: active and inactive, based on their activity levels $\mathbf{P}$. The proxy variable $\mathbf{P}$ can be any factor that the trend of the treatment effect of $\mathbf{B}$ on the outcome $\mathbf{Y}$ is altered across the stratum of $\mathbf{P}$, while is stable within each stratum of $\mathbf{P}$. We mitigate the unobserved confounding effects by approximating the stable causal effects on the outcome $\mathbf{Y}$ with an ordering on the treatments and corresponding counterfactuals. Formally, with y^b representing the probability of outcome $Y = 1$ given treatment $\mathbf{B} = b$, we define the stable counterfactual ordering as below.

myDefStable Counterfactual Ordering] Given the structural causal model in Figure 6.2, suppose within a stratification of the unobserved confounder $\mathbf{Z}$, for behaviors b_i, b_j of variable $\mathbf{B}$, if there is a stable ordering $y^{b_i} \le y^{b_j}$, we have $y^{b_i^*} \le y^{b_j}$ and $y^{b_j^*} \ge y^{b_i}$ for their corresponding counterfactuals b_i^*, b_j^*. The counterfactuals b_i^*, b_j^* are derived from applying interventions on the factual behaviors b_i, b_j, denoted by $b_i^* = b_i + \Delta_i$, $b_j^* = b_j + \Delta_j$, where Δ denotes a trivial intervention vector.

4.2.2 Counterfactual Reasoning for MBR

In MBR, user $u \in \mathscr{U}$ interacts with items $v \in \mathscr{V}$ under multiple types of behaviors $b \in \mathscr{B}$ (e.g. *click*, *favourite*, *add-to-cart* and *buy*). Here, $\mathscr{U}$, $\mathscr{V}$, and $\mathscr{B}$ denote the training set of users, items, and behaviors, respectively. Each behavior b can be either categorized into the target behavior set $\mathscr{B}_{tgt}$ or the auxiliary behavior set $\mathscr{B}_{aux}$, i.e. $\mathscr{B} = \mathscr{B}_{tgt} \cup \mathscr{B}_{aux}$. In this paper, *buy* is the only target behavior. In addition, the multi-behavior user-item interactions can be represented by a matrix $\mathbf{X} \in \mathbb{R}^{|\mathscr{U}| \times |\mathscr{V}| \times |\mathscr{B}|}$. Specifically, $x_{u,v}^{b} = 1$ if user u interact item v under the behavior type b and $x_{u,v}^{b} = 0$ otherwise.

Inspired by the advancements through the utilization of counterfactual negative samples [149, 165], we conduct counterfactual reasoning from the perspective of users' different behaviors to debias unobserved confounders. The idea of counterfactual reasoning is seeking to explore a different outcome that would have occurred if certain features had been intervened to a minimal extent. We formulate the counterfactual reasoning problem for MBR as below.

myProbCounterfactual Reasoning for MBR] Given the input of a multi-behavior interaction matrix $\mathbf{X} \in \mathbb{R}^{|\mathscr{U}| \times |\mathscr{V}| \times |\mathscr{B}|}$, user u, and item v, the recommender f outputs the index of position for item v in user u's top-K recommendation list $R_{u,K}$, the items among which is ranked by the probability that user u will interact with the item v under the target behavior "buy". For each pair of user behavior b and its counterfactual b^* which conforms to the stable counterfactual ordering in definition , we aim to have the factual and counterfactual results formulated as follows:

$$
\begin{aligned}
&f(u, v, b, \mathbf{X}|\mathbf{Z},\ \Theta) \leq K \\
&f(u, v, b^*, \mathbf{X}|\mathbf{Z},\ \Theta) > K
\end{aligned}
\tag{4.1}
$$

where $\mathbf{Z}$ is the unobserved confounder in Figure 6.2, Θ is the model parameters, K is a hyperparameter corresponding to the top-K recommendation list. A larger K implies a stricter requirement for the reversed result given the counterfactual example b^*, leading to stronger counterfactual strength.

4.3 Methodology

As demonstrated by bold and bold italics respectively in the top bar in Figure 6.3, C-MBR consists of two concurrent processes: i) User Multi-behavior Learning: this module models user preference at the intra-behavior and inter-behavior level in sequence to capture hierarchical information from multi-behavior interactions. ii) Stable Counterfactual

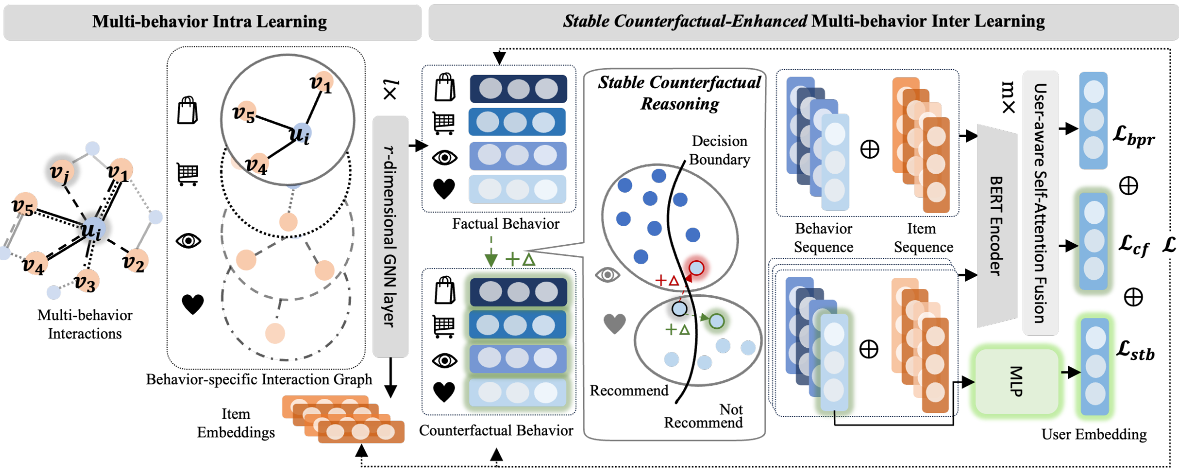


Figure 4.3: The framework of proposed C-MBR.

Reasoning: C-MBR enhances multi-behavior learning by reasoning with counterfactual examples constrained by stable counterfactual ordering in the Definition .

4.3.1 User Multi-behavior Learning

In this study, we integrate the behavior representation at the level of both intra- and inter-behavior learning. Given the multi-behavior interaction matrix $\mathbf{X}$ defined in Problem , we build behavior-specific interaction graphs $\mathcal{G}_b = (\mathcal{N}, \mathcal{E}_b)$ for $|\mathcal{B}|$ types of user behaviors, where the node set $\mathcal{N} = \mathcal{U} \cup \mathcal{V}$. Each edge $e^b_{u,v} \in \mathcal{E}_b$ exists if user u interacts item v under the behavior type b, i.e. $x^b_{u,v} = 1$ in $\mathbf{X}$, $e^b_{u,v}$ does not exist if $x^b_{u,v} = 0$.

4.3.1.1 Multi-behavior Intra Learning.

To understand users' preferences from interactions under different behaviors, we adopt the behavior modeling in [158] to model the intra-behavior relationships from each behavior-specific interaction graph $\mathcal{G}_b$. We use l multi-dimensional GNN layers to pass the messages from neighbours $\mathcal{N}^b_u$ (or $\mathcal{N}^b_v$) to the user u (or item v) and transform the neighbouring messages into r latent dimension spaces, under the consideration of vary behavior frequency. For example, the message passing to user node u at l-th layer can be represented as

$$\begin{aligned} \mathbf{H}^{b(l)}_u &= \sum_r \alpha_{b,r} \mathbf{W}_{1,r} \sum_{v \in \mathcal{N}^b_u} \mathbf{H}^{b(l-1)}_v \\ \alpha_{b,r} &= Relu(\sum_{v \in N^k_u} \mathbf{W}_2 \mathbf{H}^{b(l)}_v + \mathbf{b}_1) \end{aligned} \tag{4.2}$$

where $\mathbf{H}_u^{b(l)} \in \mathbb{R}^d$ is the hidden embeddings for behavior b of user u. $\mathcal{N}_u^b$ denotes the set of neighbouring items for user u_i. $\mathbf{W}_{1,r} \in \mathbb{R}^{d\times d}$ is a transformation matrix in the r-th dimension. $\mathbf{W}_2 \in \mathbb{R}^{|r|\times d}$ is a balancing matrix and $\mathbf{b}_1 \in \mathbb{R}^{|r|}$ is a bias. The message passing to item v is generated in the same way as for user u. We take the output of the l-th layer as the factual behavior-specific embeddings for user u and item v, denoted as $\mathbf{H}_u^b = \mathbf{H}_u^{b(l)}$ and $\mathbf{H}_v^b = \mathbf{H}_v^{b(l)}$.

The item embedding $\mathbf{H}_v \in \mathbb{R}^d$ is simply aggregated by the behavior-specific item embedding $\mathbf{H}_v^b \in \mathbb{R}^d$ with the mean function $\mathbf{H}_v = \sum_{b=1}^{|\mathcal{B}|} \mathbf{H}_v^b$

4.3.1.2 Multi-behavior Inter Learning.

After obtaining item embeddings $\mathbf{H}_v$ and behavior embeddings $\mathbf{H}_u^b$, C-MBR further learns the inter-relationship among different user behaviors by modeling the sequential information in interactions. For each user u, we chronologically concatenated the embeddings of interacted items into M subsequences (indexed by m), denoted by $\mathcal{S}_{v,m}$. Similarly, the embeddings of corresponding behaviors are also concatenated, contributing to subsequences denoted by $\mathcal{S}_{b,m}$.

$$\begin{aligned} \mathcal{S}_{v,m} &= \mathbf{H}_v^{(0)} ... \mathbf{H}_v^{(t)} ... \mathbf{H}_v^{(T)} \\ \mathcal{S}_{b,m} &= \mathbf{H}_u^{b(0)} ... \mathbf{H}_u^{b(t)} ... \mathbf{H}_u^{b(T)} \end{aligned} \tag{4.3}$$

where represents vectorwise concatenation, t denotes the t-th position in the interaction subsequence. $\mathbf{H}_v^{(t)}$, $\mathbf{H}_u^{b(t)}$ are the embeddings of corresponding item and behavior in the t-th interaction. T is a hyperparameter for the maximum length of each subsequence m. The embedding of m-th subsequence of user u is encoded by BERT [121]. Formally, $\mathbf{H}_{u,m} = BERT(\mathcal{S}_{v,m} \oplus \mathcal{S}_{b,m})$, where we concatenate the item sequence $\mathcal{S}_{v,m} \in \mathbb{R}^{T*d}$ and behavior sequence $\mathcal{S}_{b,m} \in \mathbb{R}^{T*d}$ into $\mathcal{S}_{v,m} \oplus \mathcal{S}_{b,m} \in \mathbb{R}^{T*2d}$ as the input. The elements in sequences $\mathcal{S}_{v,m}$ and $\mathcal{S}_{b,m}$ encode the *intra-behavior* relationships. $\mathbf{H}_{u,m} \in \mathbb{R}^{T\times 2d}$ is the representation of m-th sequence of user u, incorporating the *inter-behavior* relationships. In this way, the user subsequence embedding $\mathbf{H}_{u,m}$ includes both intra- and inter-behavior relationships, and sequential information of interacted items.

Then, we perform a user-aware self-attention mechanism on the embedding of each subsequence $\mathbf{H}_{u,m}$ and aggregate these M subsequences by the mean function, formally,

$$\mathbf{H}_u = \sum_{m=0}^{M} \beta_m(\mathbf{H}_{u,m}), \ where \ \beta_m = \text{softmax}(\mathbf{W}_3 \mathbf{H}_{u,m} + \mathbf{b}_2) \tag{4.4}$$

where $\mathbf{H}_u \in \mathbb{R}^d$ is the embedding for user u. $\beta_m \in \mathbb{R}^T$ is a self-attentive weight for each subsequence embedding $\mathbf{H}_{u,m}$. $\mathbf{W}_3 \in \mathbb{R}^{d\times 2d}$ is a transformation matrix. $\mathbf{b}_2 \in \mathbb{R}^d$ is a bias.

4.3.1.3 Recommendation

We calculate the probability of recommending item v to user u by the inner product of $\mathbf{H}_u$ and $\mathbf{H}_v$, denoted by recommendation score $y_{u,v}$. The Bayesian Pairwise Loss [111] is used to optimize the recommendation as below:

$$\mathscr{L}_{bpr} = -\sum_{(u,v_i,v_j)\in\mathscr{X}} ln\ \sigma(y_{u,v_i} - y_{u,v_j}) \tag{4.5}$$

where $y_{u,v} = \mathbf{H}_u^{\mathrm{T}}\mathbf{H}_v$, $\mathscr{X} = \{(u,v_i,v_j)|(u,v_i) \in \mathscr{X}^+, (u,v_j) \in \mathscr{X}^-\}$ is the training data, $\mathscr{X}^+ = \{x_{u,v}^{buy} = 1\}$ is the positive buying interactions, while $\mathscr{X}^- = \{x_{u,v}^{buy} = 0\}$ is the sampled negative buying interaction set, $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid funciton.

4.3.2 Counterfactual Reasoning

We employ stable counterfactual reasoning on MBR to enhance recommendation performance with traditional counterfactual reasoning and debias unobserved confounders with stable counterfactual examples.

4.3.2.1 Traditional Counterfactual Reasoning.

To enhance recommendation, we build counterfactual behavior through intervening on behavior-specific embedding $\mathbf{H}_u^b$ then reasoning. we define $\Delta_{u,b}$ as the personalized intervention vector applied on the behavior b of user u, representing the *minimal* change applied to $\mathbf{H}_u^b$ regarding the recommendation of item v. The intervention process can be expressed as $\mathbf{H}_u^{b^*} = \mathbf{H}_u^b + \Delta_{u,b}$, where the intervention vector $\Delta_{u,b}$ is initialized by uniformly sampling from a continuous range [-δ, 0], where ($\delta > 0$). The initialization of $\Delta_{u,b}$ with negative values is based on the intuition that removing the target item from the recommendation list may result in fewer interactions with the item, thereby contributing to a decrease in user preference. Here, we set $\delta = 1$.

Given the counterfactual embedding $\mathbf{H}_u^{b^*}$, the counterfactual recommendation score $y_{u,v}^{b^*}$ can be obtained through executing multi-behavior inter learning with $\mathbf{H}_u^{b^*}$, mathematically,

$$y_{u,v}^{b^*} = \sum_{m=0}^{M} \beta_m BERT[\mathrm{S}_{v,m} \oplus (...\mathbf{H}_u^{b^*(t)}...)]^{\mathrm{T}}\mathbf{H}_v \tag{4.6}$$

where $\mathbf{H}_u^{b^*(t)}$ is the counterfactual embeddings of the corresponding behavior in the t-th interaction. For each behavior b, we concatenate the counterfactual embedding $\mathbf{H}_u^{b^*}$ with embeddings of other behaviors $\bar{b}$ in $\mathscr{B}$ except b.

The optimization goal of counterfactual reasoning can be formulated as: suppose item v is recommended for user u in the factual top-K recommendation list ($v \in \mathscr{R}_{u,K}$), we aim to find an opposite outcome excluding the recommended item v from the top-K recommendation list ($v \notin \mathscr{R}^*_{u,K}$) via minimally intervening on each user behavior b. To be concise, the intervention is expected to be *minimal* and *effective*, mathematically,

$$\min_{\|\Delta_{u,b}\|} Rank(v|y_{u,v}) \leq K,\ Rank(v|y^*_{u,v}) > K \tag{4.7}$$

where $\|\Delta_{u,b}\|$ measure the changes caused by an intervention, $Rank(\cdot)$ function outputs the position index of item v in the ranked recommendation list given the recommendation score $y_{u,v}$. $Rank(v|y_{u,v}) \leq K$ indicates that the item v is in the top K positions in the ranked recommendation list for user u, i.e. $v \in \mathscr{R}_{u,K}$.

To solve the above optimization problem in (4.7), we model the expection of $minimal$ intervention with the the L2-normalization of $\Delta_{u,b}$, and measure the intervention *effectiveness* by moving $y^{b^*}_{u,v}$ away from $y_{u,v}$. If the second term $(y_{u,v} - y^{b^*}_{u,v})$ has a big value, there's a strong chance that v is in $\mathscr{R}_{u,K}$ but not in $\mathscr{R}^*_{u,K}$. In general, we define the following counterfactual loss $\mathscr{L}_{cf}$.

$$\mathscr{L}_{cf} = \sum_{(u,v)\in\mathscr{X}^+} \sum_{b\in\mathscr{B}} \|\Delta_{u,b}\|^2_2 - log[\sigma(y_{u,v} - y^{b^*}_{u,v})] \tag{4.8}$$

where $(u,v) \in \mathscr{X}^+$ represents the observed user-item interaction (u,v) under the target behavior buy, $\|\cdot\|^2_2$ denotes the L2-normalization.

4.3.2.2 Stable Counterfactual Reasoning.

For the reasons of proposing stable counterfactual reasoning, we consider two omissions in existing MBR models: 1) Unobserved confounding factors and 2) Buying inclination conflicts. These two omissions are described in detail in the following.

Unobserved confounding factors. MBRs are more vulnerable to unobserved confounders due to the complexity of the interactions between user behaviors and items. Existing works heavily rely on strong assumptions or additional information such as user features. We attempt to mitigate the unobserved confounding effects by leveraging information from a proxy variable, which can be modeled from multi-behavior interactions.

Buying inclination conflicts. In real-world MBR systems, interactions under different types of user behaviors mirror ascensive buying inclinations (e.g. *carting* usually shows stronger buying inclination compared to *viewing*). Table 4.1 presents the buying in-

Table 4.1: The conditional probability of buying under different interaction behaviors.

Datasets	Auxiliary Behaviors			Buying Inclination Order
	page view	*cart*	*favourite*	
IJCAI	0.33	0.37	0.15	*favourite<page view<cart<buy*
Taobao	0.12	0.15	0.11	

clination under different auxiliary behaviors in two real-world recommendation datasets. Formally, the probability $y^{b_{aux}}$ of interacted under the target behavior $b_{tgt} \in \mathscr{B}_{tgt}$ (i.e. *buy*) conditioned on an auxiliary behavior $b_{aux} \in \mathscr{B}_{aux}$ (i.e. *page view*, *cart*, *favourite*) is calculated by

$$y^{b_{aux}} = \sum_{u \in \mathscr{U}, v \in \mathscr{V}} P(x_{u,v}^{b_{tgt}} = 1 | x_{u,v}^{b_{aux}} = 1). \tag{4.9}$$

However, during the optimization of $\Delta_{u,b}$ defined in Equation (4.7), the intervention process $\mathbf{H}_u^{b^*} = \mathbf{H}_u^b + \Delta_{u,b}$ might break the ordering of buying inclinations behind different behaviors, resulting in unstable behavior counterfactual examples. In addition, pairwise attention is commonly employed in MBRs to capture various behavior contributions but neglects to buy inclinations and lacks transparency toward human comprehension.

Stable ordering loss. In this paper, we assume the presence of a proxy variable $\mathbf{Z}$ related to the unobserved confounding factors, within the strata of which the ordering of buying inclination is retained. To address the two problems above, we constrain the causal effect of counterfactual $\mathbf{H}_u^{b^*}$ on the model prediction outcome $\mathbf{Y}$ within the stable strata of this proxy variable $\mathbf{Z}$ with a stable ordering loss. Specifically, we first sort different types of user behavior $b \in \mathscr{B}$ according to the conditional probability $y^{b_{aux}}$, from least likely to most likely. For the two datasets in Table 4.1, the sorted behavior set $\mathscr{B}_s = \{$*favourite*, *page view*, *cart*, *buy*$\}$.

In case of additional influence stems from *Multi-behavior Inter Learning*, we fuse the behavior embeddings $\mathbf{H}_u^b$ of each user with a linear mapping function. For the factual space and the counterfactual intervention on behavior b, the recommendation scores $\ddot{y}_{u,v}$ and $\ddot{y}_{u,v}^{b^*}$ can be formulated as follows.

$$\begin{aligned} \ddot{y}_{u,v} &= (\mathbf{W}_4 \mathbf{H}_u^b + \mathbf{b}_3)^{\mathrm{T}} \mathbf{H}_v \\ \ddot{y}_{u,v}^{b^*} &= [\mathbf{W}_4 (\mathbf{H}_u^{b^*} \mathbf{H}_u^{\bar{b}}) + \mathbf{b}_3]^{\mathrm{T}} \mathbf{H}_v \end{aligned} \tag{4.10}$$

where $\mathbf{W}_4 \in \mathbb{R}^{|\mathscr{B}|}$ is a transformation matrix. $\mathbf{b}_3 \in \mathbb{R}^{|\mathscr{B}|}$ is a bias.

Given the sorted behavior set $\mathscr{B}_s$, for any pair of behaviors (b_i, b_j) from this set, if b_i precedes b_j, then $y^{b_i} \le y^{b_j}$. According to the stable counterfactual ordering formulated in Definition , we have $y^{b_j^*} \ge y^{b_i^*}$. Since $\ddot{y}_{u,v}^{b^*}$ denotes the counterfactual recommendation score after a negative intervention on behavior b^*, we mitigate the unobserved confounding effects via modeling the inequation with $\ddot{y}_{u,v}^{b_i{}^*} - \ddot{y}_{u,v}^{b_j{}^*}$. Therefore, we define the stability loss $\mathscr{L}_{stb}$ as follows.

$$\mathscr{L}_{stb} = \sum_{(u,v)\in\mathscr{X}^+} \sum_{(b_i,b_j)\in\mathscr{B}_s,\ i<j} -log[\sigma(\ddot{y}_{u,v}^{b_i^*} - \ddot{y}_{u,v}^{b_j^*})] \tag{4.11}$$

where $\ddot{y}_{u,v}^{b_i{}^*} - \ddot{y}_{u,v}^{b_j{}^*}$ shows the difference between the counterfactual scores when intervening on behavior b_i and b_j.

Moreover, by maximizing $(\ddot{y}_{u,v}^{b_i^*} - \ddot{y}_{u,v}^{b_j^*})$, we include the information of buying inclinations and constrain the counterfactual behavior embeddings. The buying inclination is explicitly modeled by $(\ddot{y}_{u,v} - \ddot{y}_{u,v}^{b_j{}^*}) - (\ddot{y}_{u,v} - \ddot{y}_{u,v}^{b_i{}^*})$ for any pairwise behaviors (b_i, b_j) with indexes $i < j$ in $\mathscr{B}_s$. This follows the intuition that a more important behavior b_j (ranked latter represents higher buying inclination) should drop the score $y_{u,v}$ more significantly after applying the same negative intervention vector $\Delta_{u,b}$, compared to a less important behavior b_i. We apply this stable ordering loss on a stratum of the users portioned by a stratification hyperparameter ϵ.

Finally, the objective function of C-MBR can be formulated as the sum of $\mathscr{L}_{bpr}$, $\mathscr{L}_{cf}$, and $\mathscr{L}_{stb}$.

$$\mathscr{L} = \lambda_1 \mathscr{L}_{bpr} + \lambda_2 \mathscr{L}_{cf} + \lambda_3 \mathscr{L}_{stb} \tag{4.12}$$

where λ_1, λ_2, and λ_3 are three hyper-parameters controlling the trade-off between the loss functions.

4.4 Experiments

In this paper, we conduct extensive experiments to answer the following three research questions:

- **RQ1**: How does C-MBR perform compared to the state-of-the-art (SOTA) recommendation models?
- **RQ2**: Whether the proposed stable counterfactual ordering mitigate the negative impact of unobserved confounders on recommendation performance?
- **RQ3**: How do the different loss functions in C-MBR impact the outcomes of recommendations?

4.4.1 Experimental Setups

4.4.1.1 Datasets

We evaluate C-MBR on the two public datasets **IJCAI**[1] and **Taobao**[2]. The statistics of experimented datasets are summarized in Table 4.2.

Table 4.2: Statistical information of datasets.

Datasets	#users	#items	#total interactions	#*Buy*(pct.)	#*Page View*(pct.)	#*Cart*(pct.)	# *Favourite*(pct.)
IJCAI	1637	191176	1049514	97647 (9.30%)	906394 (86.37%)	199 (0.02%)	45274 (4.31%)
Taobao	3208	180291	470139	43997 (9.36%)	383434 (81.56%)	31364 (6.67%)	11344 (2.41%)

4.4.1.2 Baseline Models.

To thoroughly demonstrate the recommendation effectiveness, we compare C-MBR with four categories of baselines: (1) **Single-behavior.** NGCF [144]. (2) **Multi-behavior intra-learning.** RGCN [113]. Since RGCN is designed for link prediction, we replace the optimization loss with the BPR loss [111]. (3) **The combination of multi-behavior intra-learning and non-sequential inter-learning.** GNMR [158] and MBGMN [160]. Besides intra-learning from behavior-specific interactions, these two models learn inter-behavior correlations implicitly with the attention mechanism and a meta prediction network, respectively. (4) **The combination of multi-behavior intra-learning and sequential inter-learning.** MBSTR [175]. Beyond the inter-correlations of different behaviors, this model includes sequential information from heterogeneous item-level interaction sequences via the transformer [136].

4.4.1.3 Evaluation Metrics and Implementation Details

We employ two common evaluation metrics: Hit Ratio (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K). C-MBR is implemented in PyTorch and optimized using the Adam optimizer, while baseline models retain their original optimizers. For IJCAI and Taobao, embedding dimensions are set as 16 and 32, respectively, with a batch size of 64. In the training, validation, and test sets, we sample 1, 99 and 99 negative items for each positive item, respectively. The learning rate is searched in [$1e^{-2}$, $5e^{-3}$, $3e^{-3}$, $1e^{-3}$, $5e^{-4}$, $3e^{-4}$].

[1]https://tianchi.aliyun.com/dataset/42
[2]https://tianchi.aliyun.com/dataset/649

Table 4.3: Recommendation performance comparison of C-MBR and selected baseline models on two real-world datasets.

	IJCAI						Taobao					
	HR@3	HR@5	HR@20	NDCG@3	NDCG@5	NDCG@20	HR@3	HR@5	HR@20	NDCG@3	NDCG@5	NDCG@20
NGCF	0.3625	0.4643	0.7014	0.2887	0.3309	0.4086	0.4432	0.5162	0.6661	0.3647	0.4148	0.4591
RGCN	0.3702	0.4404	0.6768	0.2889	0.3302	0.3986	0.4574	0.5218	0.6973	0.3652	0.4122	0.4590
GNMR	0.3781	0.4527	0.7080	0.3111	0.3490	0.4184	0.4286	0.5097	0.6608	0.3697	0.3930	0.4374
MBGMN	0.3885	0.4639	0.7159	0.3214	0.3585	0.4152	0.4243	0.5084	0.7082	0.3547	0.3894	0.4481
MBSTR	0.3745	0.4681	0.7106	0.3266	0.3595	0.4230	0.4940	0.5613	0.7150	0.4261	0.4529	0.4991
C-MBR	**0.4098**	**0.4765**	**0.7440**	**0.3706**	**0.3977**	**0.4683**	**0.5056**	**0.5944**	**0.7387**	**0.4456**	**0.4709**	**0.5156**
Improv.	5.48%	1.79%	3.93%	13.47%	10.63%	10.71%	2.35%	5.90%	3.31%	4.58%	3.97%	3.31%

4.4.2 Recommendation Performance (RQ1)

The recommendation performance comparison of C-MBR and baseline models on two datasets is reported in Table 4.3. Based on these results, we have the following observations.

C-MBR outperforms all baseline recommenders on the two real-world multi-behavior datasets. From Table 4.3, the average improvement of C-MBR to the best baseline model is 3.73% and 11.60% for HR and NDCG on the IJCAI dataset, 3.85% and 4.04% on the Taobao dataset. The improvement in terms of NDCG surpasses that of HR on both datasets. A possible reason is the $\mathscr{L}_{ord}$ learns user preferences in a more detailed manner, leading to improved ranking of candidate items. This distinction is more obvious in the IJCAI dataset than in the Taobao dataset since the order of purchase propensity across different auxiliary behaviors is more apparent in IJCAI (as shown in Table 4.1).

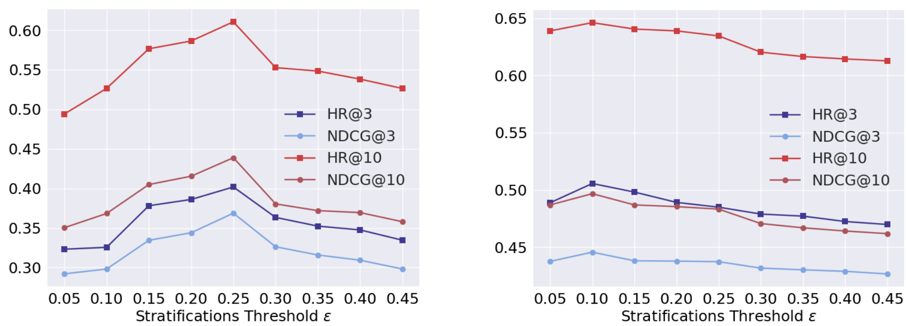
Both intra-learning and inter-learning across multiple behaviors are important. Compared to models that separately learn each behavior (i.e. NGCF, RGCN), models that learn both the intra- and inter-relationships of different behaviors (i.e. GNMR, MBGMN, MBSTR, C-MBR) generally perform better. Furthermore, MBSTR and C-MBR outperform the other baselines for they include sequential information to model the inter-relationships between users' multi-behaviors. This advantage is not evident in the Taobao dataset, probably because the interactions under each auxiliary behavior are relatively dense compared to those in the IJCAI dataset.

Counterfactual reasoning benefits the recommendation performance. Enhanced by counterfactual examples, C-MBR outperforms the best performance among other baseline (MBSTR) from 1.79% to 5.90% regarding HR, and from 3.31% to 13.47% regarding NDCG on both datasets.

4.4.3 Stratification Analysis (RQ2)

This experiment aims at justifying the motivation of stable ordering reasoning to reduce the negative impact of unobserved confounders on recommendation performance **Z** in terms of inferring the causal relationship $\mathbf{B} \to \mathbf{Y}$ in Figure 6.2.

To simplify the problem, we treat user activity as an example confounder and stratified users into active and inactive groups according to their degree of activity. Based on the stratification, we assume that the behaviors of active users who have more interactions should comply with this global ordering while forcing the interaction data that is relatively evenly distributed across behaviors of inactive users with this ordering will harm the performance. Therefore, we stratify all the users by the degree of their activity, which is estimated by the proportion of their brought items among the interacted items under all behaviors. ϵ is the stratification threshold for dividing active and inactive users.



(a) IJCAI (b) Taobao

Figure 4.4: The performance under different values of the statisfication threhold ϵ.

As illustrated in Figure 4.4, the recommendation performance for HR@3, NDCG@3, HR@10, and NDCG@10 changes based on different ϵ threshold values. For the IJCAI dataset, the peak performance is at $\epsilon = 0.25$, while for the Taobao dataset, it's at $\epsilon = 0.1$. The distinction between various ϵ values in the Taobao dataset are relatively subtle, potentially because the buying inclinations associated with different user behaviors within this dataset aren't distinctly noticeable. This performance variation indicates that the behavior ordering is maintained when properly stratifying the confounding factor, which is the degree of user activity, and this stratification enhances the overall performance.

4.4.4 Ablation Study (RQ3)

We carried out ablation experiments to evaluate how different loss functions impact the recommendation performance of C-MBR.

In Table 4.4, we evaluate the performance of three C-MBR variants with regard to HR@1(NDCG@1). The first one is **C-MBR**$\neg L_{stb}$ which is a variant of C-MBR by removing the stability loss from the joint optimization objective. The observed difference in performance between C-MBR and C-MBR$\neg L_{stb}$ on the two datasets (7.07% and 6.09%) demonstrates the effectiveness of explicitly enforcing the buying inclination of different behaviors with our proposed stable ordering. For the second variant **C-MBR**$\neg L_{cf}$, training is conducted without utilizing the counterfactual loss. As shown in Table 4.4, the omission of L_{cf} results in a reduction of 15.93% and 11.30% on the IJCAI and Taobao datasets, respectively. This shows that the performance is enhanced by capturing distinctions between factual and counterfactual user embeddings. Lastly, **C-MBR**$\neg L_{stb}, L_{cf}$ excludes both L_{cf} and L_{stb} and relies solely on training with the BPR loss [111], resulting in a performance drop of 37.81% and 25.14% for the IJCAI and Taobao datasets, respectively. This illustrates that combining counterfactual reasoning with stable ordering constraints has a beneficial synergistic impact.

Table 4.4: Ablation study w.r.t. HR@1(NDCG@1) on IJCAI and Taobao Datasets.

Variants	**IJCAI**	**Taobao**
C-MBR	0.3182	0.3647
C-MBR$\neg L_{stb}$ (Decrease)	0.2957 (7.07%)	0.3425 (6.09%)
C-MBR$\neg L_{cf}$ (Decrease)	0.2675 (15.93%)	0.3235 (11.30%)
C-MBR$\neg L_{stb}, L_{cf}$ (Decrease)	0.1979 (37.81%)	0.2730 (25.14%)

4.5 Limitations

While the proposed C-MBR method has demonstrated its effectiveness on two widely used recommendation datasets, its underlying causal assumption poses challenges for deployment in real-world recommendation systems. Specifically, establishing a stable ordering requires developers to examine the buying inclination order (as illustrated in Table 4.1) based on historical interaction statistics. This additional step introduces manual effort and may reduce scalability, particularly when dealing with large-scale or dynamically evolving platforms.

CHAPTER 5

STRUCTURED HETEROGENEITY AND UNSTRUCTURED DATA

Given the vast amount of unstructured data, models must effectively integrate both structured and unstructured information into their representations. In this chapter, we propose a novel framework that combines structured knowledge in tables with unstructured textual queries using large language models (LLMs).

5.1 Challenges in Reasoning over Structured Knowledge

With the advancement of digitalization across various industries, substantial amounts of structured knowledge are stored in tabular formats. This structured knowledge, often containing domain-specific information closely tied to different downstream tasks, complements the general knowledge acquired by Large Language Models (LLMs) during pre-training, thereby enhancing their capability to support downstream queries and reasoning [28, 122].

LLMs, leveraging their sophisticated linguistic capabilities and extensive knowledge base, have been widely utilized as one-/few-shot learners in various structured tasks [41, 43, 63, 189]. Currently, the approaches for applying LLMs on structured knowledge, including tables, fall into two primary categories: serialization-based [43, 57, 91] and operation-based methods [58, 87, 148, 169]. Serialization-based methods convert struc-

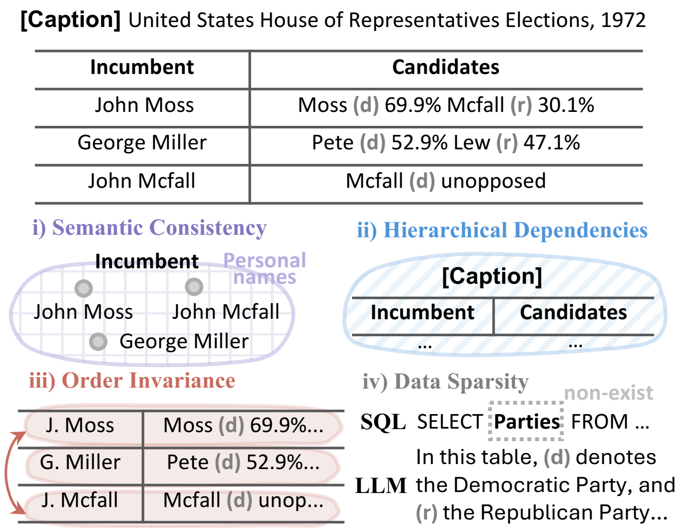


Figure 5.1: An example illustrates the three aspects of the structural relationships in tables: i) Semantic Consistency, ii) Hierarchical Dependencies, and iii) Order Invariance. Additionally, it highlights the data sparsity issue iv), where incomplete data affects SQL queries over the table .

tured knowledge into sequences of tokens, enabling the model to process the structured data in conjunction with task descriptions [91, 115]. For example, Hegselmann et al. [43] utilize a Table-to-Text model or a LLM as the serializer to convert tables into natural language strings, which are then fed into the LLM along with task descriptions. However, serializing structured data can undermine the inherent structural relationships, especially in larger tables, potentially leading to serve knowledge forgetting and diminished logical coherence during reasoning [76, 181]. Additionally, the serialized formats critically influenced the performance of LLMs [117]. The operation-based methods extract relevant information from structured data using SQL-like operations based on task requirements, and then incorporate this knowledge into LLMs to generate responses. While the SQL-like operations account for structural relationships, these methods fail to fully harness the extensive knowledge base of LLMs for effective reasoning [189]. As shown in Figure 6.1 d), the "(d)" and "(r)" represent the Democratic and Republican parties, respectively. Since the parties are not explicitly listed in a separate column, retrieving information about the political affiliations of incumbents via SQL queries is challenging. However, LLMs can easily interpret this information due to their advanced in-context learning abilities and knowledge base. Therefore, **structural relationships** and **data sparsity** are two critical challenges that current methods do not fully account for when reasoning over structured knowledge, which differs fundamentally from the unstructured text inputs that LLMs typically handle [33].

Graphs are structure-aware, making them a natural choice for modeling structural relationships. However, traditional graphs remain insufficient in effectively capturing the group relationships between rows and columns. Unlike traditional graphs, where an edge connects only two nodes, a hyperedge in a hypergraph can connect multiple cells nodes in an unordered manner. Hypergraphs consider the **structural relationships** within tabular data from three aspects: i) Semantic Consistency. Data in the cells of the same row or column in a table generally correspond to a consistent semantic category, allowing LLMs to identify and infer implicit semantic relationships. As illustrated in Figure 6.1 i), the cells in the "Incumbent" column are all personal names. ii) Hierarchical Dependencies. Hyperedges are capable of capturing intricate, higher-order dependencies within structured knowledge, such as the dependencies of the captions, headers, and cells. iii) Order Invariance. Changing word order in natural language can alter meaning, but rearranging rows or columns in a table, swapping the Moss and McFall rows in Figure 6.1 iii), does not affect the overall semantics. To address the **sparsity** issue such as the incomplete parties in Figure 6.1 iv), hypergraphs facilitate high-order information propagation between nodes and hyperedges, thereby supplementing the representations of incomplete cells with information from their neighbors. In addition, the extensive general knowledge embedded in LLMs can be leveraged to address sparse data issues.

To enhance LLMs' capabilities on structured knowledge, we propose a novel ***Hyper***graph-based ***G***eneration framework, namely , to facilitate seamless integration of knowledge from structure learning with hypergraph neural networks into LLMs, without losing focus on task-specific requirements. Specifically, explicitly guide the LLMs to augment sparse table cells with contextual information. We then construct semantics hypergraphs with the augmented table and introduce a novel Prompt-Attentive Hypergraph Learning (PHL) module that propagates task-specific inquiries in prompts along with embedded semantic knowledge across structures, and train this module jointly with the LLM. Our contributions are concluded as follows:

- **Towards structural relationship.** We propose , which uses hypergraphs to capture the semantic consistency, order invariance, and hierarchical dependencies within structured knowledge, thereby enhancing the LLM's capability to understand and reason over structured knowledge.

- **Towards data sparsity.** We design a novel hypergraph neural network to tackle the sparsity issue in tabular knowledge by utilizing the generative abilities of

LLMs and then facilitates information propagation through hyperedges.

- **Experiments.** We conduct extensive experiments on various downstream tasks involving structured data to validate the effectiveness of our proposed framework.

5.2 Problem Definition

Aiming to enhance the capability of LLMs in handling knowledge stored in structured data with hypergraphs, in this paper, we consider tables as the structured data sources to illustrate our framework. We construct a hypergraph with the structured knowledge in table $\mathcal{T}$. Each table is formally represented as $\mathcal{T} = \{o, h_i, v_{m,n} | 0 \le i \le N, 0 \le m \le M, 0 \le n \ge N\}$, where o is the table caption, h_i represents the header for the ith column, $v_{m,n}$ represents the cell at the mth row (denoted as $r_m \in \mathcal{R}$), and the nth column (denoted as $c_n \in \mathcal{C}$). As depicted in the upper left of Figure 6.3, the very upper-left cell is denoted as $v_{0,0}$. The task description prompt x, provided in natural language to the LLMs, includes a textual representation of the table $\mathcal{T}$ (in markdown format) and the essential inquiry ω regarding this table, following a specific template, $\omega \subset x$. Specifically, the essential inquiry ω can be claims in fact verification or questions in question answering. For tasks requiring the knowledge stored in $\mathcal{T}$, we aim to help pretrained LLMs (denoted as $LLM(\cdot)$) to understand and extract the structured knowledge relevant to the inquiry ω stored in $\mathcal{T}$, thereby improving the effectiveness of LLM's final generations.

5.3 Methodology

Figure 6.3 provides an overview of our proposed framework, which is designed to enhance the ability of LLMs to handle tasks that require knowledge embedded in structured data. This section details the workflow of , first augmenting the structured data with contextual information, followed by learning and integrating task-relevant structured knowledge into the LLMs to generate answers.

5.3.1 Contextual Augmentation

To address the **data sparsity** issue caused by missing or incomplete information in cells, N/A, none, or incompleted descriptions such as the example in Figure 6.1. For each table $\mathcal{T}$, we augment its caption o, columns $c \in \mathcal{C}$, and rows $r \in \mathcal{R}$ with contextual information,

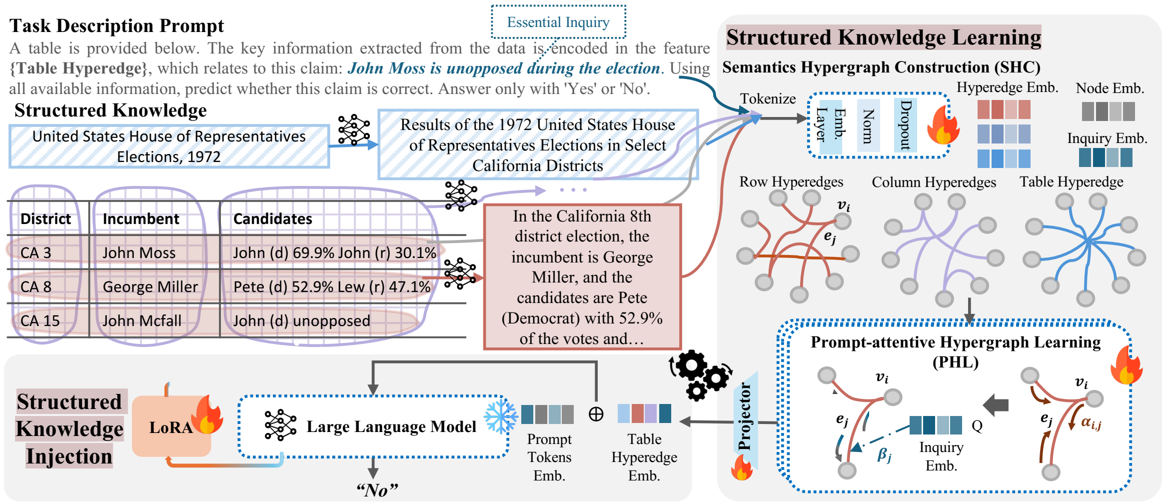


Figure 5.2: An overview of our proprosed framework.

leveraging the semantics understanding and generative ability of the large language model $LLM(\cdot)$.

Specifically, as shown in Figure 6.3, the caption o “*United States House of Representatives Elections, 1972*” is vague, as it does not specify where the elections occurred. After being supplemented with the contextual information from the table, the augmented caption, “*Results of the 1972 United States House of Representatives Elections in Select California Districts*”, denoted as $\bar{o}$, more clearly illustrates the table's content. As for the sparse cells which contain missing or incomplete data, we utilize the LLM to generate descriptions for each row and column. Formally, for the mth row,

$$\bar{r}_m = LLM(P_0(o, h_{:}, v_{m,:}) \tag{5.1}$$

where $\bar{r}_m$ represents the augmented description for the mth row containing cells $v_{m,:} = (v_{m,1}, \dots, v_{m,N})$, $h_{:} = (h_1, \dots, h_N)$ denotes the N headers of table $\mathcal{T}$, and $P_0(\cdot)$ refers to the template used to prompt the LLM in generating the corresponding augmented summary. Specifically, in this paper, we define the augmentation prompt P_0 as: “*You will be given with the table caption and headers. Please enhance the caption / describe the given row / column corresponding to the table content.*” The descriptions for the columns c_n in table $\mathcal{T}$ are generated in a similar manner, yeilding $\bar{c}_n$.

5.3.2 Structured Knowledge Learning

After augmenting the sparse data with contextual information, learns **structural relationships** over knowledge that is structurally stored through two steps: first, by constructing hypergraphs that aligns with the semantics (SHC), and second, by utilizing a novel prompt-attentive neural network for hypergraph learning (PHL). This section will elaborate on how our proposed conducts these two steps in detail.

5.3.2.1 Semantics Hypergraph Construction (SHC)

This step embeds the semantics of table $\mathcal{T}$ into a hypergraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{..., v_i, ...\}$ represents the set of node(vertex), and $\mathcal{E} = \{..., e_j, ...\}$ represents the set of hyperedges. Each hyperedge connects multiple nodes, $v_i \in \mathcal{N}_{e_j}$ denotes that the node v_i is included in the set of nodes connected by hyperedge e_j, while $e_j \in \mathcal{N}_{v_i}$ represents the hyperedge e_j is included in the set of hyperedges which connects node v_i. Each cell $v_{m,n}$ in the table $\mathcal{T}$ is treated as a node, $v_i \in \mathcal{V}$, $|\mathcal{V}| = M \times N$. The rows $r_m \in \mathcal{R}$, columns $c_n \in \mathcal{C}$, and the entire table $\mathcal{T}$ act as hyperedges, leading to three types: row hyperedges $e_{\mathcal{R}} = \{..., e_{r_m}, ...\} \subseteq \mathcal{E}$, column hyperedges $e_{\mathcal{C}} = \{..., e_{c_n}, ...\} \subseteq \mathcal{E}$, and table hyperedge $e_{\mathcal{T}} \subseteq \mathcal{E}$, $|\mathcal{E}| = M + N + 1$. The connections between the nodes $v \in \mathcal{V}$ and hyperedges $e \in \mathcal{E}$ within the hypergraph $\mathcal{G}$ are represented by an incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$, where each element $h_{i,j} = 1$ if node v_i is connected by hyperedge e_j, and $h_{i,j} = 0$ otherwise.

Specifically, we first tokenize the textual contents of cells, rows, columns, and captions using the BERT [31] tokenizer. For example, the augmented table caption is transformed into O number of tokens represented by $(\mathbf{t}_{\bar{o},1}, \mathbf{t}_{\bar{o},2}, ..., \mathbf{t}_{\bar{o},O}) = Tok_{BERT}(\bar{o})$. The tokens are subsequently passed to an embedding layer, represented as $Emb(\cdot)$, which has an output hidden dimension of d for semantics learning. Layer normalization and dropout layers are implemented in the embedding process to ensure robust generalization capabilities.

$$\mathbf{h}_{e^{\mathcal{T}}} = Dropout(LN(Emb(\mathbf{t}_{\bar{o},1}, \mathbf{t}_{\bar{o},2}, ..., \mathbf{t}_{\bar{o},O}))) \tag{5.2}$$

where $\mathbf{h}_{e^{\mathcal{T}}} \in \mathbb{R}^d$ is the hidden embedding of table hyperedge, $LN(\cdot)$ represents the layer normalization, and $Dropout(\cdot)$ refers to a dropout layer with a dropout rate 0.1. By representing the semantics of each cell content $v \in \mathcal{V}$ as node embedding $\mathbf{h}_v$ with Equation (5.2), representing the row and column descriptions $\bar{r}_m$ and $\bar{c}_n$ as row/column hyperedges $\mathbf{h}_{e^{\mathcal{R}}}$ and $\mathbf{h}_{e^{\mathcal{C}}}$, and the table hyperedge $\mathbf{h}_{e^{\mathcal{T}}}$, we construct the semantic hypergraph $\mathcal{G}$. Additionally, we also calculate the semantic embedding for the essential inquiry (as highlighted in teal-blue in the task description prompt in Figure 6.3), denoted as $\mathbf{h}_{\omega}$ for further learning.

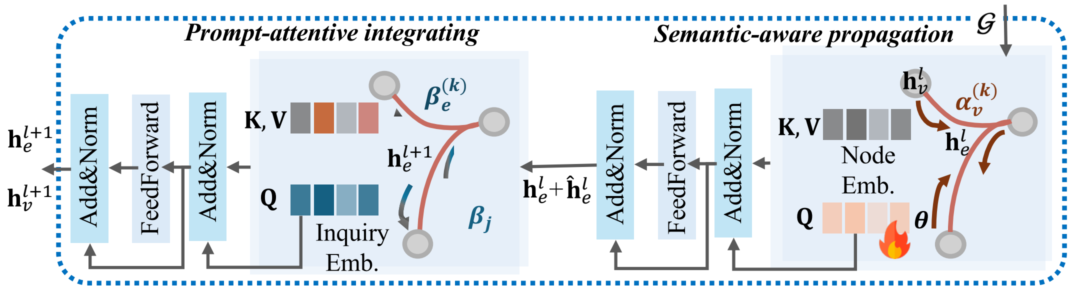


Figure 5.3: The detailed architecture of PHL.

5.3.2.2 Prompt-attentive Hypergraph Learning (PHL)

Provided with the hypergraph $\mathcal{G}$, we design a prompt-attentive hypergraph neural network to further learn structured knowledge from $\mathcal{G}$. In traditional hypergraph learning [7, 32, 34, 162], hyperedge embeddings typically do not directly participate in the propagation process; instead, hyperedges primarily serve to connect related nodes, with the focus on node embeddings. In , we aim to integrate the semantic embeddings of both nodes and hyperedges during propagation. Since the table cells contain diverse content, while the augmented hyperedge descriptions ($\bar{o}$, $\bar{r}$, and $\bar{c}$) are generated by the same LLM and maintain a consistent linguistic style, we apply node-to-edge and edge-to-node propagation using attention scores denoted by α and β with distinct designs. Specifically, inspired by [26], each PHL layer comprises two-step graph attention: first conducts *semantic-aware propagation* from nodes to their connected hyperedges, then *attentively integrate the embedding of the inquiry in the prompt* and propagates from edges to nodes.

Semantic-aware propagating. Nodes are embedded from the original table cells content with Equation (5.2), denoted as $\mathbf{h}_v$. We first propagate the original semantics embedded in $\mathbf{h}_v$ to each connected hyperedge e with the K-head hypergraph attention mechanism. The attention score $\alpha_v^{(k)}$ in node-to-edge propagation for node v at the kth head is calculated as follows.

$$\alpha_v^{(k)} = \text{L-ReLU}(\sum \mathbf{Q}^{(k)} \cdot \mathbf{K}_v^{(k)}) \tag{5.3}$$

where L-ReLU$(\cdot)$ denotes the LeakyReLU activation function, the query representation $\mathbf{Q}^{(k)} = \mathbf{W}_{k,:} \in \mathbb{R}^{1\times d}$ is the kth vector of a learnable weight $\mathbf{W} \in \mathbb{R}^{K\times d}$. We use another multi-layer perceptron to learn the key representation of the target node v, $\mathbf{K}_v^{(k)} = MLP_K^{(k)}(\mathbf{h}_v^l) \in \mathbb{R}^{1\times d}$.

Next, the information is propagated from the nodes to their connected hyperedges, formally represented as below.

$$\mathbf{h}_e^{l,(k)} = \sum_{v \in \mathcal{N}_e} \sigma(\alpha_v^{(k)}) \mathbf{V}_v^{(k)} \tag{5.4}$$

where $\mathbf{V}_v^{(k)} = MLP_V^{(k)}(\mathbf{h}_v^l) \in \mathbb{R}^{1\times d}$ denotes a multilayer perceptron used to transform the node embedding $\mathbf{h}_v^l$ to its value representation, $\mathbf{h}_e^l$ and $\mathbf{h}_v^l$ represents the hyperedge e and its connected node v, respectively, as input to the lth layer. The h_v^l is initialized by the semantics embedding of v, $\mathbf{h}_v^0 = \mathbf{h}_v \in \mathbb{R}^{1\times d}$. The softmax function $\sigma(\alpha_{v_i}^{(k)}) = \frac{exp(\alpha_{v_i}^{(k)})}{\sum_{v \in \mathcal{N}_e} exp(\alpha_v^{(k)})}$ computes the normalized attention score for each node $v_i \in \mathcal{N}_e$.The aggregation $\sum_{v\in\mathcal{N}_e}$ for the embedding of each node $\mathbf{h}_v^l$ can be performed using any aggregation function, such as summation.

As shown in Figure 5.3, to increase the representation and generation capability to be compatible with the LLM, the aggregated embedding of hyperedges $\mathbf{h}_e$ are processed using residual connections, normalization, and feed forward layers, following the architecture of transformer [136].

$$\hat{\mathbf{h}}_e^l = LN(FF(LN(\|_k \mathbf{h}_e^{l,(k)} + \mathbf{W})) + \|_k \mathbf{h}_e^{l,(k)}) \tag{5.5}$$

where $\|_k \mathbf{h}_e^{l,(k)} \in \mathbb{R}^{K\times d}$ represents the concatenation of the outputs of K heads of the multi-head attention mechanism described in Equation (5.3)(6.9). The $\hat{\mathbf{h}}_e^{l,(k)} \in \mathbb{R}^{1\times d}$ denotes the hyperedge embedding incorporating information propagated from nodes, and is then concatenated to the original semantics embedding of hyperedge e, $\mathbf{h}_e^{l+1} = \hat{\mathbf{h}}_e^l + \mathbf{h}_e$, $\mathbf{h}_e^{l+1} \in \mathbb{R}^{1\times 2d}$.

Prompt-attentive integrating. The original semantics embedding of hyperedge $\mathbf{h}_e$ are embedded from the LLM-augmented descriptions obtained with Equations (5.1)(5.2). To integrate the task requirements described in prompts into hypergraph learning, we adopt the embedding of the essential inquiry $\mathbf{h}_\omega$ in the prompt to calculate the attention score $\beta_e^{(k)}$ for edge-to-node propagation, similar to the process in Equation (5.3) (5.2). The essential inquiry can be replaced by any textual information that is of particular relevance or concern to downstream tasks in the prompts.

$$\beta_v^{(k)} = \text{L-ReLU}(\sum MLP_Q^{(k)}(\mathbf{h}_\omega \cdot MLP_K^{(k)}(\mathbf{h}_e^{l+1}))) \tag{5.6}$$

$$\mathbf{h}_v^{l,(k)} = \sum_{e \in \mathcal{N}_v} \sigma(\beta_v^{(k)} MLP_V^{(k)}(\mathbf{h}_e^{l+1})) \tag{5.7}$$

where $e \in \mathcal{N}_v$ denotes that the set of all the hyperedge e which connects the target node v, $MLP_Q^{(k)}(\mathbf{h}_\omega) \in \mathbb{R}^{1\times d}$ represents the kth vector of the query representation $MLP_Q(\mathbf{h}_\omega) \in$

$\mathbb{R}^{K\times d}$, computed by a multilayer perceptron based on $\mathbf{h}_\omega$. This operation utilizes task requirements to attentively propagate information from hyperedges to nodes. This operation intuitively aligns with the human reasoning process, where relevant rows or columns are identified first, followed by a detailed examination of individual cells when handling tasks that require structured knowledge.

Similarly, the node embedding $\mathbf{h}_v^l$ learned from multi-head attentions are further processed with the residual connections, normalization, and feed-forward layers, formally as below.

$$\mathbf{h}_v^{l+1} = LN(FF(LN({}_k\mathbf{h}_v^{l,(k)} + MLP_Q^{(k)}(\mathbf{h}_\omega))) +{}_k \mathbf{h}_v^{l,(k)})) \tag{5.8}$$

where ${}_k\mathbf{h}_v^{l,(k)}$ denotes the concatenation of the outputs of K-head attention mechanism described in Equation (5.6)(5.7).

Through the adoptions of semantic-aware propagation and inquiry-attentive integration in each PHL layer, the hypergraph neural network attains a comprehensive understanding of the hierarchical semantics embedded within structured data. This approach ensures semantic consistency, comprehensively captures hierarchical dependencies, and preserves the order invariance property of structural relationships within the knowledge structure.

5.3.3 Structured Knowledge Integration

After completing the hypergraph learning process, the knowledge linked to each cell, along with the columns, rows, and caption, is embedded in the representations of the nodes and hyperedges, which are further integrated into the generation process of LLMs.

5.3.3.1 Encoding Structured Knowledge

By connecting each cell node to the table hyperedge formed from the caption, as detailed in Section 5.3.2.1, the hidden embedding $\mathbf{h}_{e_\mathcal{T}}$ effectively captures the task-relevant structured knowledge of the entire table $\mathcal{T}$. Therefore, $\mathbf{h}_{e_\mathcal{T}}^L$ is mapped to the token space of LLM using a projector π.

$$\mathbf{e}_\mathcal{T} = \pi(\mathbf{h}_{e_\mathcal{T}}^L) \tag{5.9}$$

Here, the projected table embedding is denoted as $\mathbf{e}_\mathcal{T} \in \mathbb{R}^{d'}$, where d' denotes the dimension of the input tokens for the LLM. In , the projector π is implemented using two linear layers, with a ReLU activation function in between. The table embedding $\mathbf{e}_{\mathcal{T}}$ is then integrated into the token embeddings $\mathbf{e}_{x,:} = Tok_{LLM}(P_2(x))$ of the task description prompt x at the designated placeholder position labeled "Table Hyperedge" in natural language, as highlighted by the bold text in the upper left corner of Figure 6.3.

$$\hat{\mathbf{e}}_x = (\mathbf{e}_{x,:ph-start})(\mathbf{e}_{\mathcal{T}})(\mathbf{e}_{x,ph-end:}) \tag{5.10}$$

where $\mathbf{e}_{x,:ph-start}$ denotes all the prompt tokens preceding the placeholder, $\mathbf{e}_{x,ph-end:}$ denotes all the prompt tokens following the placeholder. Additionally, $\hat{\mathbf{e}}_x$ represents the tokens that integrate structured knowledge for further inference in the LLMs.

5.3.3.2 Training

Given the task description, markdown table, and the inquiry ω in prompt x, structured table $\mathcal{T}$, our jointly train the prompt-attentive hypergraph learning network with the LoRA [49]. The supervised fine-tuning process can be expressed in terms of the log likelihood loss. Given the input task description prompt x and target output y from the training set $\mathcal{D}$, there is,

$$\mathbb{E}_{(x,y,\mathcal{T})\in\mathcal{D}}\left[\sum_{t=1}^{T} log p_\theta(y_t|y_{1:t-1}, x, \mathcal{T})\right] \tag{5.11}$$

Here, the conditional probability distribution of the target generation output sentence y given prompt x is represented as $p_\theta(y|x) = \prod_{t=1}^{T} x_\theta(y_t|y_{<t}, x, \mathcal{T})$, where θ denotes the model parameters and T is the length of the generated sequence.

5.4 Experiments

To validate the effectiveness of , we have conduct extensive experiments to answer the following research questions.

- **RQ1**: How does the proposed perform compared to state-of-the-art (SoTA) methods when using various LLMs as backbones across different downstream tasks?
- **RQ2**: Is scalable to tables of different sizes?
- **RQ3**: How does the proposed retain the *Order Invariance* of structural relationships?

- **RQ4**: How does retain the *Semantic Consistency* and *Hierarchical Dependencies* of structural relationships?

- **RQ5**: How do the different components of contribute to to improving the performance of LLMs in learning from structured knowledge?

5.4.1 Experimental Setups

5.4.1.1 Tasks

We validate our proposed on two levels of downstream tasks that require fact-checking, and reasoning based on structured knowledge stored in tables.

- **Table Fact Verification (TFV).** This task aims at assessing the effectiveness of in fact-checking over structured knowledge. Specifically, we conduct experiments on the TabFact [151] benchmark, which contains 16k Wikipedia tables used as evidence for 118k human-annotated claims to explore fact verification with semi-structured knowledge. The ground truth answer for TFV tasks is either "yes" or "no", signifying whether the given claim is supported or contradicted by the structured knowledge stored in the corresponding table, respectively.

- **Table Question Answering (TQA).** To validate is able to facilitate LLMs to reason over structured knowledge and provide better answers to user input questions, we test on the WiKiTableQuestions [101] dataset, which includes 14,152 examples of open question-answer pairs for training and 4,344 examples for testing. The expected responses for TQA tasks are open-ended answers, which can be in the form of sentences or phrases.

Table 5.1: The statistics of training data.

Tasks	Answer Type	#Graphs	Avg. #Nodes	Avg. #Edges	Inquiry Avg. len
TFV	yes/no	1849	78.65	20.39	67.37
TQA	open answer	10141	125.11	28.94	65.05

In particular, we follow the preprocessing steps in [173] to prepare the training data, and the statistics for our final training data are listed in the table 5.1. For testing, we retain the original test sets of the two datasets, TabFact [151] and WikiTableQuestions [101], to ensure fair comparison with the selected baselines.

5.4.1.2 Baselines

We compare our proposed against 12 baseline methods, categorized by their different ways of handling tables: operation-based methods [108, 148, 170] that use external operations like SQL queries and serialization-based methods [1, 127, 178] that transform information in structures into sequences then prompt into the LLMs. In terms parameter sizes, our comparison covers a range of model sizes range from 2 billion to 70 billion parameters. For a fair comparison, we evaluate the operation-based baseline methods [108, 148, 170] using the same backbone LLMs (**LLaMA3-8B-Instruct**, **LLaMA3.2-3B-Instruct**, **Gemma-2-9B-It**, and **Gemma-2-2B-It**) as those used for our . To reduce the impact of varying instruction-following abilities among different LLMs, we adopt the instruction-tuned verions of all the selected backbone LLMs in our experiments.

Table 5.2: Comparison of the performance of our and 13 baseline methods based on varying parameter sizes, where the TFV and TQA tasks are evaluated with respect to Acc. and Denot. Acc., respectively. The first group of methods prompts LLMs with serialized tables, while the methods in each of the last four groups use the same backbone LLMs. The best and second-best results are marked with bold and underline, respectively.

Methods	Backbones	TFV			TQA			
		Acc.	Prec.	F1	Denot. Acc.	ROUGE-1	ROUGE-2	ROUGE-L
Gemma-2-2B-It [1]	-	59.80	60.55	58.54	31.88	39.68	17.81	39.60
LLaMA3.2-3B-Instruct [127]	-	54.90	56.66	54.89	24.77	35.11	16.62	35.07
TableLlama [178]	LLaMA2-7B	70.04	71.27	69.39	24.63	28.07	13.95	27.98
LLaMA3-8B-Instruct [127]	-	66.29	66.29	66.28	37.85	47.58	21.42	47.49
Gemma-2-9B-It [1]	-	75.00	75.20	74.99	46.85	55.73	24.99	55.73
Gemma-2-27B-It [1]	-	76.50	76.29	75.94	53.96	61.39	27.88	61.31
LLaMA-3.1-70B-Instruct [127]	-	**79.16**	**79.67**	**79.12**	**55.71**	**64.71**	**29.14**	**64.70**
GPT-4o-mini [98]	-	71.09	75.58	70.05	21.22	36.42	19.73	36.43
GPT-3.5-Turbo [97]	-	62.03	70.86	58.27	19.96	33.69	18.67	33.63
Text-to-SQL [108]	LLaMA3.2-3B-Instruct	57.80	58.33	56.33	28.84	35.22	12.89	34.79
Dater [170]	LLaMA3.2-3B-Instruct	60.03	58.39	59.10	33.93	39.18	13.58	39.05
CHAIN-OF-TABLE [148]	LLaMA3.2-3B-Instruct	61.09	60.49	60.49	17.14	26.81	12.97	26.67
LoRA [49]	LLaMA3.2-3B-Instruct	55.21	57.84	57.34	36.33	42.51	19.71	42.51
(Ours)	LLaMA3.2-3B-Instruct	**61.95**	**61.95**	**61.93**	**48.50**	**54.50**	**25.80**	**54.47**
Text-to-SQL [108]	LLaMA3-8B-Instruct	69.72	67.20	69.63	39.24	48.28	20.07	47.79
Dater [170]	LLaMA3-8B-Instruct	73.37	72.42	73.59	48.30	51.74	18.37	51.54
CHAIN-OF-TABLE [148]	LLaMA3-8B-Instruct	78.06	78.08	78.06	36.97	46.09	19.39	46.1
LoRA [49]	LLaMA3-8B-Instruct	66.32	67.16	63.48	49.65	56.78	25.76	56.76
(Ours)	LLaMA3-8B-Instruct	**79.14**	**80.59**	**78.95**	**55.39**	**61.45**	**27.61**	**61.37**
Text-to-SQL [108]	Gemma-2-2B-It	51.28	51.15	52.81	34.62	45.89	18.39	44.71
Dater [170]	Gemma-2-2B-It	55.68	54.82	57.55	40.95	**55.30**	19.13	**55.11**
CHAIN-OF-TABLE [148]	Gemma-2-2B-It	57.66	61.49	57.56	38.23	45.94	18.52	45.81
LoRA [49]	Gemma-2-2B-It	59.24	59.81	58.02	25.15	33.04	15.59	32.97
(Ours)	Gemma-2-2B-It	**60.64**	**61.78**	**60.64**	**41.80**	47.70	**22.01**	47.70
Text-to-SQL [108]	Gemma-2-9B-It	70.18	71.21	72.03	50.88	54.46	19.71	51.53
Dater [170]	Gemma-2-9B-It	72.88	71.60	73.31	57.94	61.95	22.91	61.64
CHAIN-OF-TABLE [148]	Gemma-2-9B-It	61.55	**79.59**	71.77	50.83	61.92	**28.16**	61.74
LoRA [49]	Gemma-2-9B-It	75.56	75.69	75.56	31.72	53.42	24.30	53.39
(Ours)	Gemma-2-9B-It	**79.14**	79.20	**79.13**	**58.54**	**62.60**	28.07	**62.56**

- **GPT-3.5-Turbo** [97] is an advanced language model from the GPT-3 family by

OpenAI, distinguished by its exceptional balance between cost-efficiency and performance, offering faster inference and lower deployment costs.

- **GPT-4o-mini** [98] is a lightweight variant of GPT-4 by OpenAI, offering strong performance with lower computational demands, ideal for resource-constrained applications.

- **LLaMA3.1-70B-Instruct** [127] is a super large model with 70 billion parameters, offering improved performance on more complicated and long-context reasoning.

- **LLaMA3-8B-Instruct** [127] is an instruction-tuned model of the LLaMA3 series with 8 billion parameters, optimized for better instruction understanding and generation.

- **LLaMA3.2-3B-Instruct** [127] is a relatively small model in the LLaMA3.2 series, fine-tuned with 3 billion parameters, and specifically designed for those tasks requiring rapid responses under limited computational resources.

- **Gemma-2-It** [1] is fine-tuned on Gemma-2 with user interactions, focusing on task-specific adaptability while ensuring efficiency through knowledge distillation from the very large model. In this paper, we use the 2B, 9B, 27B variants of Gemma-2-It.

- **TableLlama** [178] adopts LongLoRA to finetune on a dataset that includes a diverse range of serialized tables and the corresponding natural language task instructions.

- **Text-to-SQL** [108] designs in-context samples to instruct LLMs in generating SQL queries for answering questions.

- **Dater** [170] leverages LLMs to decompose the task into multiple sub-tasks, utilizing SQL queries to address each sub-task.

- **CHAIN-OF-TABLE** [148] prompts LLMs through in-context learning to iteratively produce operations and update the table, thereby constructing a reasoning chain in a structured format.

- **LoRA** [49] is a widely used technique for efficiently fine-tuning LLMs by updating a small number of low-rank weights.

5.4.1.3 Evaluation Protocol

We evaluate the generation of LLMs enhanced by our proposed framework with respect to the different tasks. For the TFV task, where the answers are either "yes" or "no", we employ **accuracy**, **precision**, **recall**, and **F1 score** as the evaluation metrics. To mitigate the impact of option bias [104, 184] in LLMs, we use a weighted version of all these metrics. For the TQA task, where responses may take the form of sentences or phrases, we adopt the following natural language evaluating metrics.

- **Denotation Accuracy (Denot. Acc.)** [102], following [58, 148], measures how closely a response matches the ground truth answer, regardless of the order of phrases in the answers.

- **ROUGE-N** measures the similarity between the LLM-generated responses and the ground truth answers by comparing overlapping n-grams, used to evaluate text summaries or translations by quantifying shared word sequences. In this paper, we report both ROUGE-1 and ROUGE-2 scores.

- **ROUGE-L** evaluates the similarity between the LLM-generated responses and the ground truth answers by identifying the longest common subsequence (LCS) and is used to assess the fluency and coherence of the generated text.

5.4.1.4 Implementations Details

For , we explore the learning rates for the LoRA module within the range of {5e-5, 1e-5, 5e-6}, while applying scaling factors for the learning rate in the PHL module (the novel hypergraph neural networks) and the projector from {1, 10, 20}. We search batch sizes from {8, 16, 32}, and conduct experiments over 1 to 4 epochs, utilizing an early stopping strategy. Specifically, for the LoRA module, we fine-tune the Query, Key, and Value projectors with a rank of 8, a LoRA alpha of 32, and a dropout rate of 0.1. For the selected baseline models, we adopt the optimal configurations from the HuggingFace[1] and accelerate inference with vllm 0.5.4[2]. All experiments in this paper are conducted on two NVIDIA A800-SXM4-80GB GPUs. For further details, please refer to our publicly released code linked in the Abstract Section.

[1]https://huggingface.co/models
[2]https://github.com/vllm-project/vllm

5.4.2 Task Performance (RQ1)

Table 5.2 presents a comparison of the performance of our with 13 baseline methods, encompassing both serialization-based methods [1, 49, 97, 98, 127, 178] and operation-based methods [108, 148, 170]. We evaluate their capabilities in structured knowledge using the TFV task on the TabFact dataset [151] and the TQA task on the WiKiTable-Question dataset [101]. In Table 5.2, the first group consists of serialized-based methods utilizing various LLMs, while the last four groups compare the performance of our with both state-of-the-art operation-based and serialized-based methods across four backbone LLMs. The following observations can be drawn from the performance results in Table 5.2.

- **Our outperforms both the operation-based and serialization-based methods based on LLMs.** It can be found in Table 5.2 that our consistently achieve competing performances across both the TFV and TQA tasks. In general, operations-based methods [108, 148, 170] achieves better outperform the methods [1, 49, 97, 98, 127, 178] of merely prompting LLMs with serialized information, even when the models have been fine-tuned on structural data [178]. This highlights the importance of maintaining structures when reasoning about questions related to structured data. Our proposed utilizes hypergraphs to encode structural information, complementing the powerful natural language capabilities of LLMs. It demonstrates an average improvements of 1.73% and 2.43% in accuracy on TFV and TQA, respectively, when compared to the second-best performances in each group. Upon reviewing the response examples, we found that CHAIN-OF-TABLE [148] encounters difficulties in TQA due to the loss of the question while reasoning over extended chains.

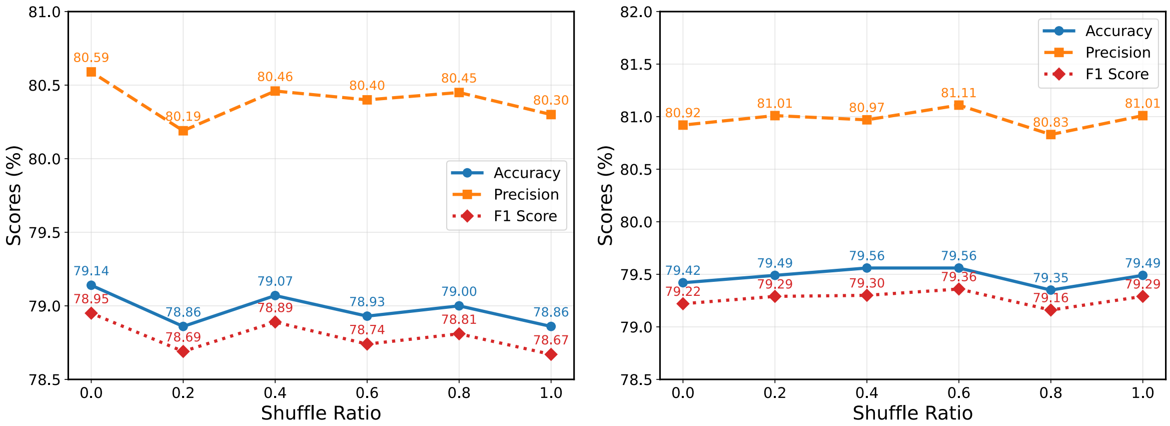
- **Our narrows the performance gap between large and small LLMs, requiring only a modest number of additional parameters.** Table 5.2 shows that instruction-tuned LLMs [1, 127, 178] with larger parameter sizes achieve better performance with serialization compared to their smaller counterparts. For instance, LLaMA-3.1-70B-Instruct achieves an accuracy of 79.16%, whereas LLaMA-3-8B-Instruct attains only 66.29%. Nevertheless, our intergrated with LLaMA-3-8B-Instruct, which adds approximately **189M** parameters (roughly one-tenth of the parameter difference between LLaMA-3-8B-Instruct and LLaMA-3.1-70B-Instruct), achieves performance comparable to the larger model across both tasks. provides average improved accuracy of 6.22% and 15.72% on the four backbone models regarding the two tasks, respectively. Similarly, based on the Gemma-2-9B-It surpasses its 27B variant by 2.64% and 2.58% in the TFV and TQA tasks with respect to accuracy, respectively. This superiority is generalizable from the TQA task to the TFV task, and is attributed to 's ability in encoding enriched structured knowledge, enabling LLMs to produce more accurate answers. Additionally, the results in Table 5.2 further demonstrate that our enhances LLMs across parameters sizes ranging from 2B to 9B.

- **Our delivers improvements in performance and training efficiency compared to other SFT methods.** As Supervised Fine-Tuning (SFT) is required in our , we compare it to the other two SFT baseline methods: LoRA [49] and TableLlama [178]. First, demonstrates significant improvements, achieving an average enhancement of 6.13% in accuracy and 15.22% in denotation accuracy [102] over LoRA for the TFV and TQA tasks, respectively. While LoRA is widely recognized as an efficient tool for instruction tuning, it proves less effective when applied to smaller-scale LLMs, such as Gemma-2-2B-It, particularly for reasoning over serialized structured data. This limitation highlights the challenges of adapting LoRA to tasks requiring nuanced structural understanding. Furthermore, when compared to TableLlama [178], which is fine-tuned on a benchmark involving serialized structured knowledge, provides a more efficient solution by fine-tuning on 189M additional parameters with very limited training data (see Table 5.1). These observations reinforce our earlier assertion that serialization-based methods can undermine the preservation of structures, further highlighting the necessity of in enhancing LLMs to fully utilize such knowledge for improved reasoning.

5.4.3 Order Invariance (RQ2)

In contrast to natural language, where changes in word order can modify the meaning of a sentence, rearranging rows or columns in a table does not affect its meaning. In , this invariance is handled with hyperedges, which represent rows and columns, are inherently unordered within the structure of hypergraphs. To assess how our proposed framework helps the LLM maintain the *Order Invariance* of structural relationships, we shuffle the rows in the test data to evaluate 's robustness to order variations.

Specifically, we randomly sampled a subset of tables from the TFV testing set and performed shuffling of the rows and columns respectively within each sampled table to introduce variability and evaluate the performance of our proposed . Figure 5.4 displays



(a) Shuffle Rows (b) Shuffle Columns

Figure 5.4: Performances of under different variances of order simulated by shuffling.

the performances of which uses LLaMA3-8B-Instruct as the backbone model, across different shuffle ratios. The x-axis represents the sampling ratio, while the y-axis indicates performance scores with respect to accuracy, precision, and F1 score. As decipted in Figure 5.4, framework demonstrates stable performance despite variations in row and column order. The accuracy variance of 0.0109 for row shuffling and 0.00043 for columns shuffling, respectively. This stability underscores the robustness of in maintaining structural representation integrity from the perspective of order invariance, thereby validating our previously stated rationale for employing hypergraphs.

5.4.4 Semantic Consistency and Hierarchical Dependencies (RQ3)

Beyond quantitative metrics, we also conduct qualitative evaluations to investigate whether retains semantic consistency and hierarchical dependencies of structural relationships during reasoning. In Figure 6.6, we randomly selected two cases and visualized the attention weights between each cell node and the hyperedges associated with the claim's content. This visualization provides insights into how prioritizes and propagates information between table elements and their relevance to the given queries/claims, specifically demonstrating its ability to maintain semantic consistency and hierarchical dependencies.

The semantic consistency in structural realtionships suggests that cells in the same column are similar in semantics. In `Case 1`, the claim pretains to the class of a team named "evansville triplets". first augmented each row with easy-to-understand natural

Table 5.3: The ablation study results of using LLaMA3-8B-Instruct as the base model on the TFV task. Red signifies degradation in percentage.

Methods	**TFV**			
	%Acc.	**%F1**	**%Prec.**	**%Recall**
(Ours)	77.20 0.00	77.46 0.00	79.98 0.00	77.20 0.00
w/o PHL	70.96 ↓6.24	70.86 ↓6.60	71.35 ↓8.63	70.96 ↓6.24
w/o PHL, w/ HGNN	72.70 ↓4.50	72.54 ↓4.92	73.03 ↓6.95	72.70 ↓4.50
w/o Inquiry Emb.	72.63 ↓4.57	73.39 ↓4.07	74.22 ↓5.76	74.22 ↓2.98

language descriptions based on the row context, as shown in the bottom right of Figure 6.6. This augmentation enables to better interpret cells with specific missing values ("none" in the "Class" columns), leading to similar weights for these cells and their counterparts within the same column. In `Case 2`, the claim queries about the highest crowd participation, which requires examining the "Crowd" column to identify the largest number. Though the crowd numbers for these teams vary, the weights assigned to the column hyperedges are similar thanks to the augmented column descriptions (omitted here) and the Semantics Hypergraph Construction (SHC) in . This is more evident in the weight matrix associated with the table hyperedge shown in the right bottom of 6.6 (b). Even though the venue names are quite different, the cells in the "Venue" column share similar weights.

The Hierarchical Dependencies refers to the hierarchy across cells, columns, rows, and the whole table. As depicted in Figure 6.6, the attention of is primarily focused on the cells and the rows/columns related to the claim. This focus extends from cell nodes to row/colum hyperedges, and then the table hyperedges, gradually diminishing in intensity. For example, in Figure 6.6 (b), the weights assigned to the queried cell "Junction oval", the evidence cell "33100", and the relevant column "Crowd" exceed the average weights of 0.27 and 0.16 in the weight matrices corresponding to the column hyperedges and the table hyperedge, respectively. This demonstrates how the attention mechanism spans the hierarchical structure, emphasizing specific elements within the table.

5.4.5 Ablation Study (RQ4)

We are also curious about the contribution of each component in contributes to the enhancements of . As shown in Table 5.3, we successively removed the proposed prompt-attentive hypergraph learning (PHL) module, substituted the PHL module with HGNN [34], and removed the LLM-based argumentation. Note that this ablation study is conducted under hyperparameters setting different from those used for the results in Table 5.2.

It can be observed from the experimental results in Table 5.3 that the original framework of our proposed delivers the best performance on verifying the factual knowledge stored in structured data. Firstly, removing the PHL module and directing the semantic embeddings directly to the projector results in a 6.24% reduction in accuracy. Furthermore, to examine the role of hypergraph neural networks in enhancing LLMs' comprehension of structured knowledge, we replace our proposed PHL module with the classical HGNN [34], leading to a 4.50% decrease in performance compared to , as shown in the third row of Table 5.3. This performance degradation highlights the effectiveness of hypergraphs in representing structured knowledge. Specifically, We attribute this decline to the inability of HGNN to adequately leverage the information encoded in hyperedges for node updates during propagation. Additionally, we explore the impact of incorporating the inquiry embedding in the PHL module. As demonstrated in the last row of Table 5.3, removing the inquiry embedding causes a substantial 5.76% drop in precision and a more moderate 2.98% decline in recall. This suggests that incorporating inquiry embeddings helps LLMs mitigate the bias toward over-generating positive responses, fostering more cautious reasoning by integrating the essential inquiry within prompts when processing structured data.

5.4.6 Scalability (RQ5)

Scalability is a critical concern, as large tables pose significant challenges for LLMs, which often struggle to interpret and integrate context from lengthy prompts [82, 171]. To evaluate the performances of on tables of varying sizes, we divided the testing tables in [151] into three classes: **small** (#rows ≤ 5 and #columns≤ 5), **medium** (6≤#rows≤10 and 6≤#columns≤ 10), and **large** (#rows≥ 10 and #columns≥ 10). We compare the performance of LLaMA3-8B-Instruct [127], Dater [170], and , with LLaMA3-8B-Instruct serving as the backbone model for all.

As illustrated in Figure 5.6, with the table sizes increases there are generally declines in all of the models. demonstrates relatively stable performance in precision with a variance of 8.84, surpasses LLaMA3-8B-Instruct of 10.87. For small tables, surpasses Dater in precision and recall by 6.39% and 7.42%, respectively. However, Dater demonstrates superior recall performance compared to for medium and large tables. Upon careful examination and analysis of the true positive cases, we found that this is primarily due to the LLMs in Dater being inclined to generate positive answers. This is also benefits from its effective approach of decomposing queries and tables into sub- questions and sub-tables. The declines of in larger tables are primarily observed in recall. We attribute this to the limited number of hypergraph learning layers and aim to address this issue in the future through more sophisticated graph techniques.

5.5 Limitations

The primary limitation of the proposed HyperG framework lies in its scalability. To further investigate this aspect, we conduct experiments on HyperG using tables of varying lengths in Sec. 5.4.6. The results show that, compared with Dater, a more flexible framework designed to enhance the reasoning ability of LLMs on tabular data, HyperG demonstrates slightly lower scalability despite achieving consistently superior performance in accuracy and reasoning quality. This limitation, however, is not unique to our approach but rather a common challenge among graph neural network–based methods due to their inherent computational and memory complexity. Future work could explore more efficient graph construction strategies, parameter-sharing mechanisms, or hybrid architectures to improve scalability while preserving HyperG's strong reasoning capability.

[Caption] bosse field

[Augmented Caption] Overview of Evansville baseball teams at Bosse Field

Team	Sport	League	Played	Class	Championships
...	...	...	...	...	...
evansville braves	baseball	three-I league	1946-1957	b	three - i league title 1946 , 1948 , 1956 , 1957
evansville white sox	baseball	southern league	1966-1968	none	none
evansville triplets	baseball	american association	1970-1984	aaa	american association title 1972 , 1975 , 1979
evansville otters	baseball	frontier league	1995-present	indp	frontier league title 2006

[Claim] The evansville triplet baseball team be in the aaa class.

Team	Sport	League	Played	Class	Championships
...	...	...	...	...	...
0.21	0.20	0.21	0.20	0.20	0.21
0.24	0.19	0.20	0.20	0.29	0.23
0.36	0.32	0.34	0.31	0.35	0.36
0.26	0.22	0.25	0.24	0.29	0.26

cell nodes → row hyperedges

[3rd Row Description] The row corresponds to the Evansville White Sox, a baseball team that played in the Southern League from 1966 to 1968 and had a missing class, with no championships won.

[4th Row Description] The row corresponds to the Evansville Triplets, a baseball team that played in the American Association from 1970 to 1984 and won three championships in 1972, 1975, and 1979.

(a) Case 1. Information about the baseball teams at Bosse Field.

[Caption] 1961 VFL season

Home Team	Away Team	Venue	Crowd	Date
Hawthorn	North Melbourne	Glenferrie oval	14000	5 August 1961
Essendon	Geelong	Windy hill	27500	5 August 1961
Collingwood	Footscary	Victoria part	16889	5 August 1961
Carlton	South Melbourne	Prince park	16889	5 August 1961
Sk kilda	Melbourne	Junction oval	33100	5 August 1961
Richmond	Fitzroy	Punt road oval	15547	5 August 1961

[Claim] during the 1961 VFL season , junction oval venue record the highest crowd participation

Home	Away	Venue	Crowd	Date
0.12	0.13	0.14	0.33	0.30
0.14	0.16	0.16	0.50	0.41
0.14	0.17	0.26	0.51	0.49
0.14	0.16	0.25	0.50	0.45
0.19	0.29	0.33	0.55	0.49
0.13	0.15	0.20	0.24	0.24

cell nodes → column hyperedges

Home	Away	Venue	Crowd	Date
0.14	0.14	0.13	0.15	0.15
0.16	0.15	0.15	0.19	0.16
0.16	0.16	0.15	0.16	0.16
0.16	0.17	0.19	0.18	0.24
0.22	0.20	0.22	0.24	0.25
0.15	0.15	0.15	0.16	0.17

cell nodes → the table hyperedge

(b) Case 2. The results of the 1961 Victorian Football League (VFL).

Figure 5.5: Visualization of the weights between cell nodes and different hyperedges in two random cases.

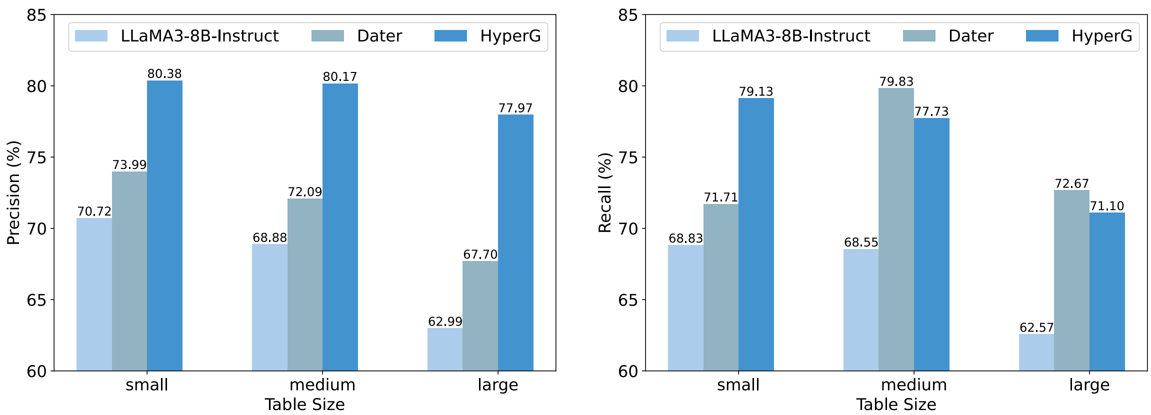


Figure 5.6: Performances of on tables of different sizes, w.r.t. precision (left) and recall (right)

CHAPTER 6

DYNAMIC EVOLUTION ADAPTION

Beyond the challenge of heterogeneous dependencies in static scenarios, representations in deep models should also be capable of capturing dynamic evolutions over time and context. A typical scenario sensitive to such temporal and contextual changes is news recommendation. The following sections elaborate on our solution, which leverages representations from various deep models.

6.1 Challenges in News Recommendations

In an era of information overload, news recommender systems (NRSs) play a vital role in delivering personalized content, enabling users to discover timely and relevant news amid the overwhelming volume of information. To deliver timely and personalized recommendations, NRSs are required to accurately model time-variant factors that capture evolving trends and user interests, while also accounting for time-invariant factors, such as the semantics of exposed news. This dual focus ensures that users receive updates on trending topics that align with their preferences, thereby enhancing the effectiveness of recommendations.

Time is indispensable in news recommendation, as it is crucial for ensuring that users receive the most current and relevant information. Multiple temporal factors, whether observed or not, can influence users' satisfaction with recommended items [109]. Attempting to account for all of these factors is impractical and would add considerable complexity [70]. Given the constraints on data availability, such as the lack of contextual user information due to privacy concerns, the recency and popularity of news items are widely used as two key time-variant factors in recommendations [106, 141]. As illustrated in 6.1 (a), news users are influenced by the recency and/or popularity of news items. The bold black arrow on the far left of the timeline indicates the release time of news items `A` and `B`. As time progresses from left to right, the recency of news `A` and `B` diminishes. The popularity of items diminishes as the color transitions from darker to lighter shades. In the example shown in Figure 6.1, the first user prefers fresh news, while the other prioritizes the popularity of news. Existing NRSs can be categorized by the time-variant factors they consider: recency [29, 119], popularity [6, 106], and both [35, 50, 131]. The first category of models addresses time-decaying correlations between exposed news and their recency from the user's perspective by distinguishing between short-term and long-term user preferences, leveraging fixed or adaptive time windows to prioritize recently interacted news while down-weighting outdated content [119]. Popularity-based models identify correlations between global popularity and exposed news by analyzing topic prevalence and prioritizing news within trending topics [6]. Methods that account for both recency and popularity consider the correlations of exposed news by incorporating these two factors [50, 167]. For example, in the user-item interaction graph in [50], recency of an item node can be encoded by node features, while the degree of the item node shows its popularity.

While these methods effectively approach time, they primarily rely on correlations and are, therefore, skewed by redundant associations. A more recent study [25] examines the causal impact of time on exposed news but simplifies time by treating it as a static variable. This approach does not capture the complex dependencies between multiple time-variant factors and overlooks how their causal influences on the user satisfactions evolve over time. The causal influences of time-variant factors on user satisfactions are time-variant. As shown in Figure 6.1 (b), we consider the example of a user driven by popularity. If there is no recommendation at time t_1, the user would prefer news `A` (denoted by blue strip in Figure 6.1) about NVIDIA stocks at time t_2, possibly because he is a stakeholder of NVIDIA. However, if news `B` (denoted by orange strip) about the United States chip export controls is presented to the user at time t_1, he might sell his NVIDIA stock by time t_2 and thus lose interest in news `A` (denoted by blue strip), even if it is popular then. Conversely, if news `C` (denoted by green strip) about United States policy is shown to and clicked by the user earlier, sparking their interest in policy, they would prefer news `B` (denoted by orange strip) about chip export controls at time t_2 despite its lack of popularity. A similar situation holds for recency-driven users. Therefore, the

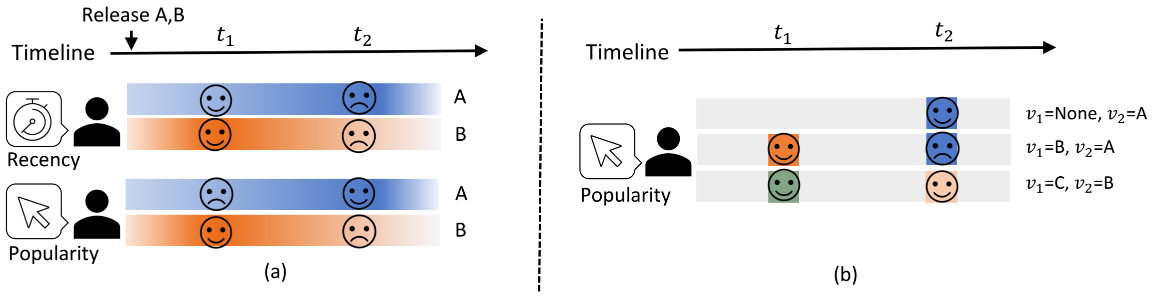


Figure 6.1: Toy examples of (a) recency- and popularity-driven users, where positions further to the left indicate higher recency, and darker positions indicate higher popularity; (b) how the causal influence of exposed items affect future user satisfactions, where v_1, v_2 represents news items exposed at time t_1 and t_2, respectively.

causal impact of time-variant factors on user satisfaction evolves over time, necessitating the modeling of causal relationships in an evolving manner.

Effectively modeling the evolving causal effects of past exposures on future user satisfaction requires separately capturing time-variant and time-invariant covariates, as these two types of covariates provide different types of user preference signals while also introducing potential confounding effects. In this paper, we contend that these two types of covariates should be modeled independently. *Time-variant covariates*, consisting of various time-variant factors, risk overemphasizing recent trends or highly popular content if integrated into user modeling based on correlations. Separately modeling time-variant covariates allows for more precise modeling of the causal relationships among these time-variant factors and their confounding effects on the treated news and outcoming user satisfactions. Similarly, by isolating *time-invariant covariates*, recommenders can more accurately represent inherent user preferences independent of time, mitigating the impact of fluctuating noise and confounding effects arising from redundant associations. Therefore, separately modeling covariates in news recommendations from time-variant and time-invariant perspectives enables a more nuanced handling of confounding effects.

In this work, we approach the news recommendation task from a causal perspective, carefully addressing both time-variant and time-invariant covariates to accurately model the evolving causal influences of exposed news items on users' satisfaction. Previous sequential recommenders effectively captured information from sequences of historically interacted items. However, these models primarily rely on correlations between time-variant factors and interacted items, rather than explicitly learning the causal relationships between them over time, separate from time-invariant factors. To explicitly model the evolving causal relationships of time-variant covariates, we utilize a transformer-based causal block to incorporate news popularity and recency. Simultaneously, we extract users' inherent preferences from extensive textual information using Large Language Models (LLMs), treating these preferences as the positive component of time-invariant covariates. In general, we design a **CA**u**S**al **T**ime-aware **Rec**ommender, named **CAST-Rec**, aiming to make recommendations based on the estimation of user satisfaction by modeling the evolving causal influences of news exposures. Although our primary focus is on news popularity and recency, CAST-Rec is adaptable and can seamlessly incorporate other time-variant factors that may be more prominent in specific scenarios.

We conclude our contributions as follows.

- To understand the causal influence of recommendations on user satisfaction over time, we are the first to consider the evolving causal influence of news exposures.
- To effectively model complex dependencies among time-variant factors, we design a transformer-based causal block to address the time-variant covariates.
- To carefully capture time-invariant covariates, we utilize the semantic understanding, generalization, and generation capabilities of Large Language Models.
- To validate the effectiveness of our proposed CAST-Rec, we conduct extensive experiments on two real-world news recommendation datasets.

6.2 Task Formulation

We denote a user $u \in \mathscr{U}$ and a news item $v \in \mathscr{V}$, where $\mathscr{U}$ and $\mathscr{V}$ are the set of users and items, respectively. In this paper, we re-scrutinize the task of news recommendations from a causal view. For each user u, the exposed item and the corresponding user satisfaction are treated as treatment and outcomes and denoted as variable $\mathbf{V}$ and variable $\mathbf{S}$, respectively. To represent the evolution of time, we define a set of timeslices $\mathscr{T} = \{\mathbf{T}_i \in \mathbb{R}\}$, the past time period is denoted as $\mathbf{T}_{1:t} = \{\mathbf{T}_1, ..., \mathbf{T}_t\}$, and the prediction time is denoted as $\mathbf{T}_{t+1}$. In this paper, we use subscripts to denote time. For example, the news item exposed at time $\mathbf{T}_i$ is represented as v_i, the sequence of news item exposed in the past is denoted as $\mathbf{V}_{1:t}$. To model the evolving causal influence of $\mathbf{V}$ on $\mathbf{S}$, we consider both time-variant covariates $\mathbf{C}$ and time-invariant covariates $\bar{\mathbf{C}}$. For time-variant covariates $\mathbf{C}$, c_i generally refers to the value of any time-variant covariate at time $\mathbf{T}_i$. In this paper, we incorporate two time-variant factors as time-variant covariates: news popularity $\mathbf{P}$

and recency $\mathbf{R}$. Popularity and recency during the past time period $\mathbf{T}_{1:t}$ are indicated as $\mathbf{P}_{1:t} = \{p_i^v | v \in \mathbf{V}_{1:t}, 1 \leq i \leq t\}$ and $\mathbf{R}_{1:t} = \{r_i^v | v \in \mathbf{V}_{1:t}, 1 \leq i \leq t\}$, where p_i^v denotes the popularity of the exposed news item v at time $\mathbf{T}_i$, r_i^v denotes the corresponding recency. For time-invariant covariates $\bar{\mathbf{C}}$, we learn $\bar{c}$ for the user u from long-term browsing history and/or demographic information.

6.2.1 News Recommendation from a Causal View.

News recommendation methods predict users' preferences on candidate news items by analyzing their browsing history, where time is crucial. To capture the evolving causal relationship between exposed news items and user satisfaction while accounting for time, we consider both the causal influences of time-variant covariates and time-invariant covariates.

6.2.1.1 Covariates and Confounders.

To clearly articulate our causal model, we firstly introduce covariates and confounders, with a particular focus on their relationships and distinctions.

A *covariate* is an endogenous variable that is observed in the scope of causal study and correlated with both the treatment variable and/or the outcome variable. In a causal study, covariates can serve as confounders, predictors, mediators, or nuisance variables, but only confounders directly affect the validity of causal inference. Covariates do not always introduce bias but provide additional information, and can therefore be included in predictions.

A *confounder* is a specific type of covariate that simultaneously influences both the treatment and the outcome variables while not affecting the causal pathway between them as mediator. In causal inference, a confounder introduces bias into the estimation of causal effects because it builds spurious associations between the treatment and outcome variables. Therefore, additional techniques, such as backdoor adjustment or stratification, are required to carefully address confounders and ensure accurate causal estimation.

Previous studies have either solely addressed confounding effects [14, 25] or treated temporal factors as additional information [35, 50, 106, 131]. In this paper, we consider both the additional information and the confounding effects introduced by time-variant covariates $\mathbf{C}$ and time-invariant covariates $\bar{\mathbf{C}}$. Taking the recency of browsed news $\mathbf{R}$ as an example time-variant covariate, it can improve the system's accuracy by capturing nuanced user preferences related to the additional temporal information. However, recency **R** also introduce confounding effects by simultaneously influencing item exposures **V** and users' satisfaction **S**, as users may remain continuously focused on recently occurring news events.

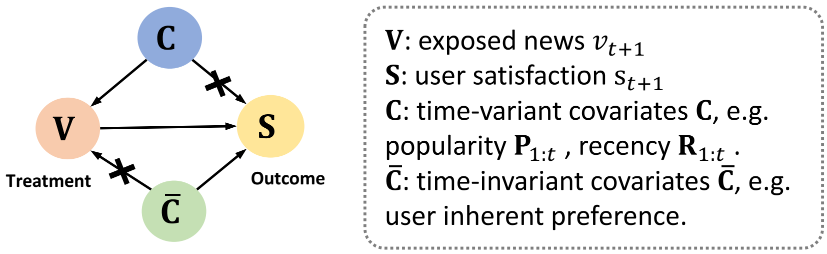


Figure 6.2: A structural causal model illustrating the *confounding effects* of covariates in news recommendations. Exposed news item v_{t+1} at prediction time $\mathbf{T}_{t+1}$ is treated as the treatment and user's satisfaction s_{t+1} on the item is treated as the outcome. This causal model considers both the confounding effects sourced from the time-variant covariates $\mathbf{C}$ for the past time period $\mathbf{T}_{1:t}$, and the confounding effects sourced from the time-invariant covariates $\bar{\mathbf{C}}$. As indicated by the crosses, we address the confounding effects from the time-variant covariate $\mathbf{C}$ and the time-invariant covariate $\bar{\mathbf{C}}$ via mitigating the information flow of paths $\mathbf{C} \rightarrow \mathbf{S}$ and $\bar{\mathbf{C}} \rightarrow \mathbf{V}$, respectively.

6.2.1.2 Structural Causal Model

To address the confounding effects from both time-variant and time-invariant covariates, we depict the causal relationships in the scenario of news recommendations with a structural causal model. In the structural causal model depicted in Figure 6.2, the exposed news v_{t+1} at time $\mathbf{T}_{t+1}$ is considered as the treatment variable, while the corresponding user satisfaction **S** as the outcome representing how much the user likes the exposed news. Figure 6.2 also represents the two types of confounding effects sourced from time-variant covariates **C** and time-invariant covariates $\bar{\mathbf{C}}$, respectively. Directed arrows in Figure 6.2 depict the information flow from the cause factor to the effect factor, indicating that the child node is influenced by the ancestor node. Each edge is depicted as follows.

- $\mathbf{V} \rightarrow \mathbf{S}$ shows user satisfaction with the news item recommended by the system.
- $\mathbf{C} \rightarrow \mathbf{V}$ shows how time-variant covariates, such as the popularity and recency of correlated news items, influence the next news exposed to the user.

- $\mathbf{C} \rightarrow \mathbf{S}$ indicates the evolving nature of user preferences, showing how time-variant covariates influence user satisfaction with exposed items.
- $\bar{\mathbf{C}} \rightarrow \mathbf{V}$ indicates user's time-invariant preferences which can be learned from the user's browsing history and/or demographic profile, will implicitly influence whether the news item will be exposed to the user by the system.
- $\bar{\mathbf{C}} \rightarrow \mathbf{S}$ shows that the user's time-invariant preferences which can be learned from browsing history and/or demographic profile affects their satisfaction towards the exposed news.

From a causal view, we aim at calculating the next recommended news item by predicting user satisfaction via estimating the causal influence of path $\mathbf{V} \rightarrow \mathbf{S}$. To this end, we have to carefully address the two types of confounding effects introduced by time-variant and time-invariant confounders to effectively leverage the benefits of these covariates while mitigating the drawbacks.

6.2.2 Confounding Effects on Recommendation

In Figure 6.2, we categorize the confounding effects of these three variables into two types: time-variant and time-invariant. This section will elaborate on how we account for and address these two types of confounder and their adverse effects on recommendations.

6.2.2.1 Time-variant confounding effect

In this paper, we consider two time-variant covariates: news popularity $\mathbf{P}$ and recency $\mathbf{R}$. The popularity and recency change over time and can be derived from covariates in past user interactions. For example, the recency of a news v browsed at time $\mathbf{T}_i$ can be calculated as $\mathbf{T}_i - \mathbf{T}_v$, where $\mathbf{T}_v$ denotes the release time of the news article v or the first time the news v appears in the system, $\mathbf{T}_v \leq \mathbf{T}_i$. The time-variant covariates $\mathbf{C} = \{\mathbf{P}, \mathbf{R}\}$ concurrently influence both the exposed news treatment $\mathbf{V}$ and the outcome user satisfaction $\mathbf{S}$. Specifically, at time $\mathbf{T}_i \in \mathbf{T}_{1:t}$, substituting $\mathbf{C}$ by $\mathbf{P}$, the path $\mathbf{C} \rightarrow \mathbf{V}$ in Figure 6.2 indicates that when the system selects exposed news v, it considers the news popularity p_i^v at time $\mathbf{T}_i$; the path $\mathbf{C} \rightarrow \mathbf{S}$ shows that users' satisfaction for this news v can be influenced by its current popularity p_i^v, where generally popular news is more likely to gain favor. A similar situation holds for the recency variable $\mathbf{R}$.

The confounding effects of time-variant covariates $\mathbf{C}$ negatively influence the estimation of the causal relationship between $\mathbf{V}$ and $\mathbf{S}$ because it provides a backdoor path

$\mathbf{V} \leftarrow \mathbf{C} \rightarrow \mathbf{S}$. Treating item popularity $\mathbf{P}$ as a time-variant covariates, this confounding effect is also known as popularity bias in recommender systems. For example, political news is rarely viewed by the youth, who are more active on the news platform compared to the middle-aged demographic. However, the middle-aged, who have an interest in political news, may not be exposed to these news items if the system is largely influenced by popularity. In this paper, we initially apply time-aware stratification to separate these time-variant covariates, i.e., news popularity $\mathbf{P}$ and recency $\mathbf{P}$, from the interacted news items, then address the confounding by blocking the backdoor path $\mathbf{C} \rightarrow \mathbf{S}$ using the CAST-Rec framework. The rationale of time-aware stratification will be detailed in Section 6.3.1, and our proposed CAST-Rec will be introduced in Section 6.3.3.

6.2.2.2 Time-invariant confounding effect.

As for the time-invariant covariate $\bar{\mathbf{C}}$, we account for the news content extracted from browsing history or user demographic profiles. Though these time-invariant features bring benefits to the prediction, they also bring confounding effects by concurrently influencing both the exposed news treatment $\mathbf{V}$ and the outcome user satisfaction $\mathbf{S}$ as shown in Figure 6.2. Specifically, the path $\bar{\mathbf{C}} \rightarrow \mathbf{V}$ illustrates that the system's exposures implicitly mirrors time-invariant user inherent preferences; concurrently, these time-invariant preferences $\bar{\mathbf{C}}$ also explicitly reflected in the satisfactions $\mathbf{S}$, represented by path $\mathbf{F} \rightarrow \mathbf{S}$.

Similarly, these time-invariant features from either browsing history or demographic profile $\bar{\mathbf{C}}$ introduce confounding effects through a backdoor path $\mathbf{V} \leftarrow \bar{\mathbf{C}} \rightarrow \mathbf{S}$. To carefully leverage the time-invariant covariate, we extract and incorporate the textual information from extensive browsing sequences into our predictions, while blocking the implicit influences of the path $\bar{\mathbf{C}} \rightarrow \mathbf{V}$ to circumvent the confounding. In this paper, the proposed CAST-Rec extracts and models the information flow through the path $\bar{\mathbf{C}} \rightarrow \mathbf{V}$ with LLMs, as elaborated in Section 6.3.3.

6.2.3 Problem Formulation

Based on the structural causal model depicted in Figure 6.2, we formulate the problem we focus in our proposed CAST-Rec for news recommendations as below. [News Recommendations] For each user u, we aim to learn a news recommendation model g that is capable of predicting user satisfaction score $\mathbf{s}_{t+1}$ based on estimating the causal effects of exposed news items v_{t+1} at prediction time $\mathbf{T}_{t+1}$.

$$\mathbf{s}_{t+1} = g(v_{t+1}|\bar{\mathbf{C}}, \mathbf{V}_{1:t}, \mathbf{P}_{1:t}, \mathbf{R}_{1:t}) \tag{6.1}$$

where $\mathbf{V}_{1:t}$ includes the historical browsed news items, $\mathbf{P}_{1:t}$ represents the corresponding popularity, $\mathbf{R}_{1:t}$ involves the recencies during the past time period $\mathbf{T}_{1:t}$. The extracted preferences of user u acting as time-invariant covariate are represented as $\bar{\mathbf{C}}$.

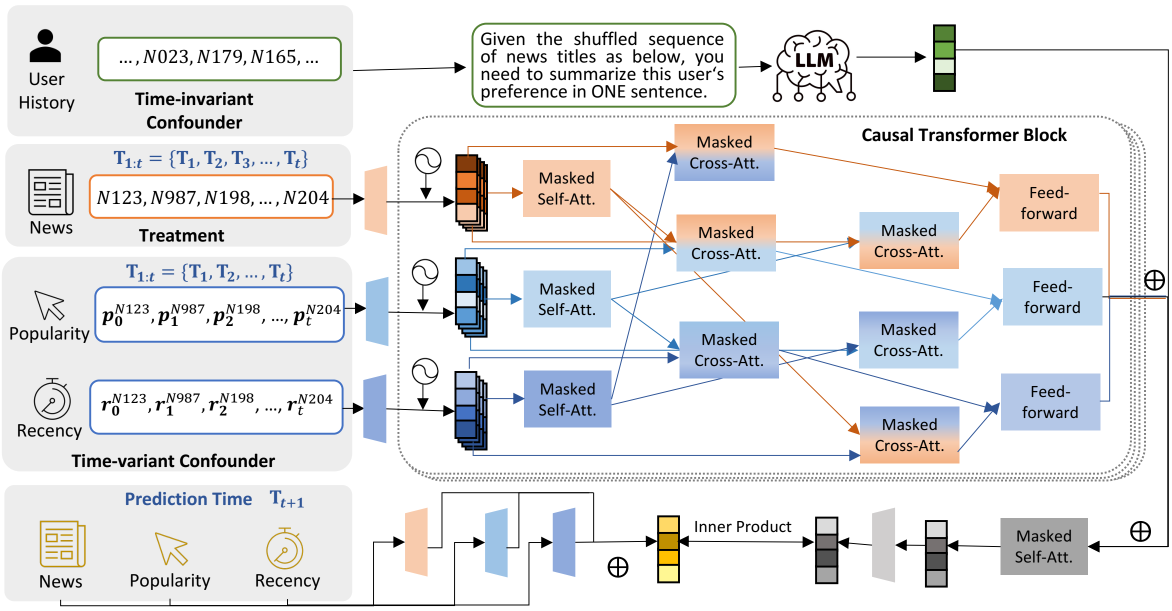


Figure 6.3: Overview of our proposed CAST-Rec framework.

6.3 Methodology

We propose CAST-Rec to predict user's satisfactions on exposed news by estimating the evolving causal relationship between them, taking into account both time-variant and time-invariant covariates. Before learning the evolving influences, we first prove the rationality of independently modeling the exposed news and the time-variant covariates (news popularity and recency) in Section 6.3.1. Then, as shown in Figure 6.3, CAST-Rec carefully leverage the information sourced from time-invariant covariates with LLMs (Section 6.3.2) and time-variant covariates with a series of transformer-based causal blocks (Section 6.3.3), respectively.

6.3.1 Causality Preparation: Time-aware Stratification

Before learning, we carefully investigate the rationality of independently modeling the treatments variable (exposed news) and the time-variant covariates (time-variant factors such as recency and popularity). Inspired by previous works on causality [8, 103], we employ a commonly-used stratification method to retrieve the independence between the treatment news **V** and time-variant covariates **C** in Figure 6.2, allowing for separate modeling of the covariates and the treatment. This section firstly introduces the rationale of stratification method in causality, then theoretically prove the independence of **CV** based on the structural causal model in Figure 6.2.

6.3.1.1 Stratification in Causality.

The idea of stratification is to divide the data into two or more distinct strata (subgroups) such that the confounder **C** and **V** are independent of each other in each subgroup. A key challenge of applying stratification is how to divide users to ensure that within each subgroup, the independence **CV** always exists. To resolve this, we divide subgroups by numerous time slices, the time intervals of which tend to be infinitesimal. Within each subgroup with time slices $\mathbf{T}_i$, the time-variant covariates can be regarded as static. Specifically, based on the casual model shown in Figure 6.2, the probability of the exposure of news item at time $\mathbf{T}_i$ can be presented as $P(\mathbf{V}=1|\mathbf{C}=c_i,\bar{\mathbf{C}})$. At time $\mathbf{T}_i$, the user either be exposed with news item or not; only the satisfaction **S** of the users who are exposed to the news is taken into account. However, for the recommendations in a past time slices $\mathbf{T}_i \in \mathbf{T}_{1:t}$, the partition of users who are exposed with the news item is determined, yielding the independence of **CV**. The following subsections proves the effectiveness of this time-aware stratification. For simplicity, we omit v and the time-invariant variable $\bar{\mathbf{C}}$ for the probability $P(\mathbf{V}=1|c_i,\bar{\mathbf{C}})$.

6.3.1.2 Proof of independence

We aim to prove the independence within each subgroup that $\mathbf{T}_i \in \mathbf{T}_{1:t}$ in order to separately model the time-variant covariate variables (i.e., popularity and recency of items). With any subgroup $\mathbf{T}_i$ and the corresponding time-variant covariate $\mathbf{C}=c_i$, each user u has either been exposed to the news v (divided into partition $u \in \mathscr{U}_1$) or not being exposed to the news v (divided into partition $u \in \mathscr{U}_2$). For simplicity, we omitted the notation v of news. Formally, the two partitions of users can be presented as,

$$\begin{aligned}
&\mathscr{U}_1 = \{u \in \mathscr{U} | P(\mathbf{V}=1|c_i,\ u) = 1 = P(\mathbf{V}=1|u)\}\ and \\
&\mathscr{U}_2 = \{u \in \mathscr{U} | P(\mathbf{V}=0|c_i,\ u) = 1 = P(\mathbf{V}=1|u)\}\ , \qquad (6.2) \\
&\qquad\qquad\qquad \mathscr{U}_1 \cup \mathscr{U}_2 = \mathscr{U}.
\end{aligned}$$

where $P(\mathbf{V}=1|c_i,u)$ represents the probability of news item $\mathbf{V}$ has been exposed to user u given the time-variant covariate $\mathbf{C}=c_i$. Intuitively, the user variable $\mathbf{U}$ is independent of all the variables in Figure 6.2. Given the causal graph in Figure 6.2, within the subgroup of $\mathbf{T}_i$, the probability of $\mathbf{V}=1$ can be represented as below.

$$\begin{aligned}
&P(\mathbf{V}=1|\mathbf{C}=c_i) \\
&\quad (a) \sum_{u\in\mathscr{U}} P(\mathbf{V}=1|\mathbf{C}=c_i,\mathbf{U}=u)P(\mathbf{U}=u|\mathbf{C}=c_i) \\
&\quad (b) \sum_{u\in\mathscr{U}_1} P(\mathbf{V}=1|\mathbf{U}=u)P(\mathbf{U}=u|\mathbf{C}=c_i) \\
&\quad + \sum_{u\in\mathscr{U}_2} P(\mathbf{V}=1|\mathbf{U}=u)P(\mathbf{U}=u|\mathbf{C}=c_i) \\
&\quad (c) \sum_{u\in\mathscr{U}_1} P(\mathbf{V}=1|\mathbf{U}=u)P(\mathbf{U}=u|\mathbf{C}=c_i) \\
&\quad (d) \sum_{u\in\mathscr{U}_1} P(\mathbf{V}=1|\mathbf{U}=u)P(\mathbf{U}=u) = P(\mathbf{V}=1)
\end{aligned}$$

where Equation (a) uses the law of total probability by conditioning on all possible users $\mathbf{U}$, Equation (b) and Equation (c) are based on Eq.(6.2), Equation (d) is based on the independence $\mathbf{UC}$. Therefore, the time-variant covariate $\mathbf{C}$ is independent from the news exposure $\mathbf{V}$. Given $\mathbf{CV} \mid \mathbf{U}$, we are able to separately model the covariate $\mathbf{C}$ and exposed items $\mathbf{V}$. In this paper, we account for both item popularity and recency for time-variant covariates, $\mathbf{C} = \{\mathbf{P},\mathbf{R}\}$. A detailed illustration of how we leverage these time-variant covariates will be given in Section 6.3.3.

6.3.2 LLM for Time-invariant Covariates

Considering the confounding effects of backdoor path $\mathbf{V} \leftarrow \bar{\mathbf{C}} \rightarrow \mathbf{S}$, CAST-Rec utilizes LLMs to extract the essential information flow of $\bar{\mathbf{C}} \rightarrow \mathbf{S}$ from historical browsing sequences, while mitigating the path $\bar{\mathbf{C}} \rightarrow \mathbf{V}$. This is motivated by the following three key capabilities of LLMs: 1) The powerful **semantics understanding capability** of LLMs empowers them to uncover nuanced user interests by interpreting deeper meanings embedded within content. For instance, when analyzing news articles, LLMs can go beyond understanding the explicit semantic meaning to capturing subtle, implicit emotions, such as the humor or hilarity associated with mentioning a specific comedian's name. This ability to detect both surface-level information and hidden emotional cues enables LLMs to deliver more personalized and contextually aware recommendations. Traditional methods, such as using language models like BERT, may overlook these nuances and fail to capture inherent user preferences effectively during user modeling. 2) The **generalization capability** of LLMs is harnessed to abstract away minor variations, generating a meaningful time-invariant representation based on users' historical interactions. For example, while the content of a news article remains static, underlying elements like writing style may subtly evolve over time. These changes, though time-varying, are too subtle to justify modeling explicitly as time-variant variables. In contrast, traditional approaches like content-aware attention may inadvertently incorporate these subtle time-varying features, leading to potential noise and reduced clarity in the resulting representations. 3) The **generative capability** of LLMs enhances transparency by producing human-readable summaries of previously browsed news. By distilling complex data or lengthy content into concise summaries, LLMs provide users with a clearer understanding of the underlying information, making it more accessible and actionable. Specifically, these summaries allow users to directly observe how the system models their inherent interests in news articles, independent of temporal factors such as writing style or popularity. In contrast, traditional methods including attention mechanisms often lack the ability to explicitly express how user interests are represented, making their decision-making process less interpretable and transparent.

The employment of LLMs here contributes in a two-fold manner: 1) **mitigating the path $\bar{\mathbf{C}} \rightarrow \mathbf{V}$**: The time-varying nature of the historical browsing sequence $\mathbf{V}_{1:t}$ arises from its ordering based on interaction time. Therefore, directly encoding this sequence as a time-invariant covariate $\bar{\mathbf{C}}$ using language models is inaccurate. However, when browsing news items are shuffled, the advanced generative capabilities of LLMs allow them to identify and isolate inherent preferences that remain consistent over time. 2) **enhancing the path $\bar{\mathbf{C}} \rightarrow \mathbf{S}$**: The purpose of encoding historical browsing contents in news recommendations is to capture inherent user preference from the textual data [106]. LLMs are capable of understanding complex language patterns and semantics from vast amounts of textual information, allowing them to provide highly contextualized and concise summaries. These diverse yet concise summaries generated by the LLMs helps capture the nuances of user preferences, thereby enhancing the modeling of $\bar{\mathbf{C}} \rightarrow \mathbf{S}$.

Specifically, to capture user time-invariant inherent preferences from the browsing history, there are intuitively two approaches: 1) encoding the news content with language models first and then concatenating it as a sequence, or 2) concatenating first and then encoding. This approach aims to incorporate the user's interactions with various news items over time, providing a comprehensive view of their interests. The modeling of sequential information in the first method heavily depends on the performance of language models and can harm the semantic coherence between two consecutively browsed news items. Therefore, we adopt the second approach, first concatenating the textual feature f^v (e.g., title, abstract, category of news), of each news item v in chronological order. In CAST-Rec, we adopt the title for its balance between being informative and concise. The textual information of news features browsed over past time period $\mathbf{T}_{1:t}$ can be represented as $f^{v_1}f^{v_2}...f^{v_t}$, where v_t represents the news item interacted with at time t. Furthermore, given that an extensive browsing history for

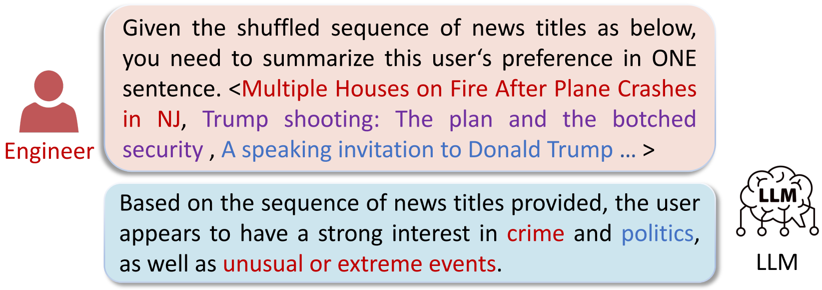


Figure 6.4: An example prompt in a chat with LLMs.

active users introduces redundant relationships, we aim for user interests to be concise and consistent over time. The concatenation $f^{v_1}f^{v_2}...f^{v_t}$ of news contents is formatted with a prompting template $Template(\cdot)$ as shown in Figure 6.4, which instructs the LLM to summarize the essence of user interests with one sentence. This process yields a formatted personalized history $Prompt = Template(f^{v_1}f^{v_2}...f^{v_t})$. By providing a clear and structured prompt, the LLM can effectively generate concise and relevant summaries that capture the essential aspects of the user's interests. The box in the upper left corner of Figure 6.3 shows an example of a prompt. Then, this personalized history $Prompts$ is tokenized into the input sequence of tokens $X = (x_1, x_2, x_3, ..., x_m)$, where m is the length of input sequence. The LLM generates an output sequence, we take the last hidden states of this generation as the representation of time-invariant user interest $\bar{c}$. For a model with l number of layer $L(\cdot)$, we can have the time-invariant user representation as follows.

$$\mathbf{e}^{\bar{C}} = \mathbf{H}_{LLM}^{(l)} = L^{(l)}(h_1^{(l-1)}, h_2^{(l-1)}, ..., h_m^{(l-1)}),\ \mathbf{H}_{LLM}^{(l)} = (h_1^{(l)}, h_2^{(l)}, ..., h_m^{(l)}) \tag{6.3}$$

where $\mathbf{H}_{LLM}$ represents the hidden states of the LLM, $h_i^{(l)} \in \mathbb{R}^{\frac{d_1}{n}}$ is the embedding of i-th token at the l-th layer, and $\bar{c} \in \mathbb{R}^{d_1}$ is the representation of user interest at the last layer, d_1 is the dimensionality of the output from the LLM model.

This time-invariant user interest $\bar{c}$ can be decoded into an explicit one-sentence user interest summarized from the browsing history. Since only news titles are used along with the prompt instructions, and without inputting any temporal information, the output $\bar{c}$ is expected to capture the essential user interests, which are consistent over time.

6.3.3 Time-aware Modeling for Time-variant Covariates

Based on the time stratification preparation in Section 6.3.1, we are capable of separately modeling the treated exposed news and potential time-variant covariates. This section consider two time-variant covariates, news popularity $p \in \mathbf{P}$ and recency $r \in \mathbf{R}$, and conduct time-aware modeling to carefully leverage these two time-variant covariates.

6.3.3.1 Popularity

The CTR of a news item can be calculated at four different granularities: news category, news subcategory, the news item itself, and words in the news title. For example, the CTR based on news category represents the probability of a click occurring on news items within the same category as the exposed item v, given that news items in the same category are exposed.

$$p_i^v = \frac{1}{|\mathcal{U}_1^{v'}||\mathcal{V}^{cat_v}|} \sum_{u \in \mathcal{U}_1^{v'}} \sum_{v' \in \mathcal{V}^{cat_v}} P(\mathbf{S}=1|u,v',\mathbf{T}_i) \tag{6.4}$$

where $u \in \mathcal{U}_1^{v'} \subset \mathcal{U}_1$ represents the set of users exposed to news item v', $v' \in \mathcal{V}^{cat_v}$ represents the set of news items in the same category of the exposed item v, the conditional probability $P(\mathbf{S}=1|u,v',\mathbf{T}_i)$ represents the probability of user u clicks given the exposure of item v' at time $\mathbf{T}_i$.

6.3.3.2 Recency

The recency of news is calculated by the reciprocal of the difference between current time t and the time that it was first exposed in the system. Similar to the popularity, the recency also experimented with the four different granularities. Formally, the category-level recency of news item v at time $\mathbf{T}_i$ can be represented as follows, denoted by $r_i^v \in \mathbb{R}$.

$$r_i^v = min(0,\ min_{v' \in \mathcal{V}^{cat_v}}(\mathbf{T}_i - \mathbf{T}_{v'})) \tag{6.5}$$

where $\mathbf{T}_{v'}$ denotes the first time that news $v' \in \mathcal{V}^{cat_v}$ is exposed to any user $u \in \mathcal{U}$ in the system.

6.3.3.3 Multi-input Causal Block

As time-variant covariates, the news popularity and recency provide invaluable information for making predictions. To this end, we employ a transformer-based multi-input causal block to model the evolving causal influence between these covariates and exposed news items, aiming to improve the prediction of user satisfaction.

Inputs. For each past time slice $\mathbf{T}_i \in \mathbf{T}_{1:t}$, three time-variant variables are treated as the inputs: (i) treatment variable exposed news v_i, corresponding (ii) covariate variable news popularity p_i, and (iii) covariate variable news recency r_i. For simplicity, we omit the notation of v for the popularity and recency here. Given the differing dimensions of these three time-variant variables, each input is projected through its respective multi-perception layer before being processed by the causal block layers. Note that the embeddings for exposed news are initialized using Xavier uniform initialization. For instance, exposed news items arranged in chronological sequence $\mathbf{V_{1:t}} = \{v_1, v_2, ..., v_t\}$ are formally projected as the following formulation shown.

$$\mathbf{H}_0^{\mathrm{V}} = MLP^{\mathrm{v}}(v_1 v_2 ...\ v_t) \tag{6.6}$$

where $\mathbf{H}_0^{\mathrm{V}} \in \mathbb{R}^{d_2 \times t}$ is the hidden representation of the series of treatments at time $\mathbf{T}_{1:t}$, d_2 is the dimensionality of hidden representation in CAST-Rec. The projection network is a multi-layer perceptron MLP^{V} consisting of two linear layers with the Tanh activation function in between. Similarly, the news popularity $\mathbf{P}_{1:t} \in \mathbb{R}^t$ and the news recency $\mathbf{R}_{1:t} \in \mathbb{R}^t$ are projected using MLP^{P} and MLP^{R}, respectively, yielding $\mathbf{H}_0^{\mathrm{P}}$ and $\mathbf{H}_0^{\mathrm{R}}$. Both MLP^{P} and MLP^{R} consist of two linear layers with an ELU activation function in between. This difference arises because the ELU activation function is particularly effective for positive inputs. This is the case for popularity and recency as formulated in Equations (6.5)(6.4). In contrast, the Tanh activation function helps to normalize the embedding of exposed news.

Self-/Cross-Attention. The initially hidden representations $\mathbf{H}_0^{\mathrm{V}}$, $\mathbf{H}_0^{\mathrm{P}}$, $\mathbf{H}_0^{\mathrm{R}}$ for exposed news, popularity, and recency are then inputted into the transformer-based causal block. Specifically, they are first fed into multiple attention heads, each requiring a set of query, key, and value vectors, denoted as $Q^{(n)}, K^{(n)}, V^{(n)} \in \mathbb{R}^{T \times \frac{d_2}{N}}$, respectively. The attention heads are indexed by n, and there are N such heads in total. For any initial hidden representation $\mathbf{H}_0 \in \{\mathbf{H}_0^{\mathrm{V}}, \mathbf{H}_0^{\mathrm{P}}, \mathbf{H}_0^{\mathrm{R}}\}$,

$$\begin{aligned} Q^{(n)} &= \mathbf{H}_0 W_Q^{(n)} + \mathbf{1} b_Q^{(n)\top} \\ K^{(n)} &= \mathbf{H}_0 W_K^{(n)} + \mathbf{1} b_K^{(n)\top} \\ V^{(n)} &= \mathbf{H}_0 W_V^{(n)} + \mathbf{1} b_V^{(n)\top} \end{aligned} \tag{6.7}$$

where $W_Q^{(n)}$, $W_K^{(n)}$, $W_V^{(n)} \in \mathbb{R}^{d_2 \times \frac{d_2}{N}}$ represents weights for the i-th self-attention head, $b_Q^{(n)}$, $b_K^{(n)}$, $b_V^{(n)} \in \mathbb{R}^{\frac{d_2}{N}}$ are the biases for query, key, values, respectively, $\mathbf{1} \in \mathbb{R}^{\frac{d_2}{N}}$ is the vector of ones. The attention values in the i-th head are formally computed as follows.

$$Attn^{(n)}(Q^{(n)}, K^{(n)}, V^{(n)}) = softmax(\frac{Q^{(n)} K^{(n)\top}}{\sqrt{\frac{d}{N}}}) V^{(n)} \tag{6.8}$$

The outputs of these N attention heads are then concatenated.

$$MHA(Q, K, V) = (Attn^{(0)}, \ldots, Attn^{(n)}, \ldots, Attn^{(N)}) \tag{6.9}$$

We first compute the self-attention representation $\dot{\mathbf{H}}$. By substituting the hidden representation $\mathbf{H}_0$ in Equation (6.7) with $\mathbf{H}_0^{\mathrm{V}}$, we can calculate the respective query, key, and values Q^{V}, K^{V}, V^{V}. For simplicity, we omitted the index i for multiple heads. Consequently, the self-attention representation $\dot{\mathbf{H}}^{\mathrm{V},(l)}$ at the l-th causal block is computed as follows.

$$\dot{\mathbf{H}}^{\mathrm{V},(l)} = SelfAtt(\mathbf{H}^{\mathrm{V}}) \tag{6.10}$$

$$SelfAtt(\mathbf{H}^{\mathrm{V}}) = LN(FF(MHA^{(l)}(Q^{\mathrm{V}}, K^{\mathrm{V}}, V^{\mathrm{V}})) + \dot{\mathbf{H}}^{\mathrm{V},(l-1)}) \tag{6.11}$$

Here, $MHA^{(l)}$ represents the multi-head attention in Equation (6.9) in the l-th blocks, the feed forward network is denotes as $FF(\cdot) = Linear(ReLU(Linear(\cdot)))$, and $LN(\cdot)$ stands for linear normalization. To avoid over-fitting, the final linear projection layer present in the original transformer decoder [136] is omitted. Similarly, the variables $\mathbf{P}$ and $\mathbf{R}$ of news popularity and recency are also self-attentively embedded into $\dot{\mathbf{H}}^{\mathrm{P},(l)}$ and $\dot{\mathbf{H}}^{\mathrm{R},(l)}$, respectively. Calculating self-attention allows the model to better understand intra-variable evolving dependencies.

Next, we compute the pairwise cross-attentions $\tilde{\mathbf{H}}$ to capture inter-variable evolving dependencies. Given self-attention outputs $\dot{\mathbf{H}}$ from Equation (6.11), the cross-attention can be calculated from both directions: taking the cross-attention between $\mathbf{V}$ and $\mathbf{P}$ as an example, using $\dot{\mathbf{H}}^{\mathrm{V}}$ as the query and $\dot{\mathbf{H}}^{\mathrm{P}}$ as the key and values, and vice versa. Calculating cross-attention in both directions allows the models to fully capture the evolving dependencies over time. Specifically, beyond the causal influence represented by $\mathbf{V}_i \to \mathbf{P}_i$ at time $\mathbf{T}_i$, the exposure v_i will also influences the news popularity at time $p_{i+1}^{v_i}$, denoted by $\mathbf{V}_i \to \mathbf{P}_{i+1}$.

$$\begin{aligned} \tilde{\mathbf{H}}^{\mathrm{V,P}(l)} &= CrossAtt(\dot{\mathbf{H}}^{\mathrm{V}}, \dot{\mathbf{H}}^{\mathrm{P}}), \\ \tilde{\mathbf{H}}^{\mathrm{P,V}(l)} &= CrossAtt(\dot{\mathbf{H}}^{\mathrm{P}}, \dot{\mathbf{H}}^{\mathrm{V}}) \end{aligned} \tag{6.12}$$

$$CrossAtt(\dot{\mathbf{H}}^{\mathrm{V}}, \dot{\mathbf{H}}^{\mathrm{P}}) = LN(FF(MHA^{(l)}(Q^{\mathrm{V}}, K^{\mathrm{P}}, V^{\mathrm{P}})) + \mathbf{H}^{\mathrm{V}}))) \tag{6.13}$$

where $CrossAtt(\dot{\mathbf{H}}^{\mathrm{P}}, \dot{\mathbf{H}}^{\mathrm{V}})$ is calculated in the similar way as $CrossAtt(\dot{\mathbf{H}}^{\mathrm{V}}, \dot{\mathbf{H}}^{\mathrm{P}})$. This process helps the model understand how the covariates influence the exposure of news items over time. By learning these dependencies, the model can make more accurate predictions based on the combined effects of news exposures $\mathbf{V}$, popularity $\mathbf{P}$, and recency $\mathbf{R}$.

After computing the cross-attention, a feed-forward layer is adopted to combine all the relative information of a variable, for example,

$$\mathbf{H}^{\mathrm{V}} = FF(\tilde{\mathbf{H}}^{\mathrm{V,P}}, \tilde{\mathbf{H}}^{\mathrm{P,V}}) \tag{6.14}$$

Finally, after L multi-input causal blocks, the final output will be self-attentive aggregated into a time-variant user representation.

$$\mathbf{e}^{\mathrm{C}} = SelfAtt(\mathbf{H}^{\mathrm{V}}\mathbf{H}^{\mathrm{P}}\mathbf{H}^{\mathrm{R}}) \tag{6.15}$$

Position Encoding. In the original transformer model [136], positional encoding is used to preserve the order of hidden states within a sequence, as the attention mechanism tends to overlook order information. This position encoding is crucial in news recommendations to ensure that the temporal sequence of news exposures is accurately captured.

In our multi-input causal block, we employ relative positional encoding. This choice is made because absolute time information is already captured in the representations of news popularity and recency, which are calculated based on timestamps. Relative positional encoding allows us to focus on the relative time distances between news items, which is essential for modeling the evolving influence of earlier news exposures on later interactions. Considering the news items at position $\mathbf{T}_i$ and $\mathbf{T}_j$ for any $\mathbf{T}_i \le \mathbf{T}_j$, we have the following modifications to the key and value in the attention calculation.

$$\begin{aligned} \alpha^K_{j,i} &= W^K_{i-j} \\ \alpha^V_{j,i} &= W^V_{i-j} \end{aligned} \tag{6.16}$$

where W^K_{j-i}, and $W^V_{j-i} \in \mathbb{R}^{t\times \frac{d}{N}}$ are two relative positional encoding matrices applied to the keys and values in the attention mechanism, respectively. Then, the scores $\alpha^K_{j,i}$ and $\alpha^V_{j,i}$ are used to adjust the attention score $Attn(Q,K,V)_j$ from Equation (6.8).

$$\begin{gathered} Attn(Q,K,V)_j = \sum_{i\le j} softmax_i(\frac{Q_j K'_i}{\sqrt{\frac{d}{N}}})V'_i, \\ K'_i = K_i + \alpha^V_{j,i}, \;\; V'_i = V_i + \alpha^V_{j,i} \end{gathered} \tag{6.17}$$

where $softmax_i$ calculates relative to position $\mathbf{T}_j$. Note that the latter position $\mathbf{T}_j$ only participates in the attention calculation for past positions $\mathbf{T}_i$ or itself.

6.3.4 Recommendation

Lastly, CAST-Rec makes predictions by integrating the causal influences from the exposed items and the two covariates, via three information flows $\mathbf{C} \to \mathbf{S}$, $\mathbf{V} \to \mathbf{S}$, and $\bar{\mathbf{C}} \to \mathbf{S}$, respectively.

Objective. At prediction time $\mathbf{T}_{t+1}$, we calculate the representation of any candidate news item $\hat{v}$ for exposure based on its popularity $p^{\hat{v}}_{t+1}$, recency $r^{\hat{v}}_{t+1}$.

$$\mathbf{e}^{\hat{v}} = MLP(MLP^{\mathrm{V}}(\hat{v})MLP^{\mathrm{P}}(p^{\hat{v}}_{t+1})MLP^{\mathrm{R}}(r^{\hat{v}}_{t+1})) \tag{6.18}$$

where we use a $MLP(\cdot)$ that includes two linear layers with ELU activation function to project the candidate news item into the matching space. Then, we estimate the user satisfaction $\mathbf{s}_{t+1}$,

$$\mathbf{s}_{t+1} = (\mathbf{e}^{\bar{\mathrm{C}}}\mathbf{e}^{\mathrm{C}})^\top \cdot \mathbf{e}^{\hat{v}} \tag{6.19}$$

We trained our CAST-Rec model using the Bayesian Personalized Ranking (BPR) loss [111], which aims to maximize the difference in the user satisfaction scores between positive and negative candidate items. Formally,

$$\mathcal{L}_{BPR} = -\frac{1}{|\mathcal{O}|}\sum_{(v,v')\in\mathcal{O}} ln\ \sigma(\mathbf{s}^v - \mathbf{s}^{v'}) \tag{6.20}$$

where $\mathbf{s}^v$ and $\mathbf{s}^{v'}$ are the predicted user satisfaction scores for news items v and v', respectively. $\mathscr{O} = \{(v, v')|v \in \mathscr{O}^+, v' \in \mathscr{O}^-\}$ is the training data, $\mathscr{O}^+$ involves the positive interacted news items, while $\mathscr{O}^-$ is the sampled negative interacted news items, $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

Interpretability. Regarding the interpretability of the recommendation process, CAST-Rec introduces the following advancements: First, it applies causal inference to uncover more interpretable variable relationships compared to correlation-based methods. Second, it accounts for confounding effects from both time-variant and time-invariant covariates, reducing spurious correlations that introduce bias and improving model transparency. Third, CAST-Rec leverages LLMs to generate content that explicitly reflects users' time-invariant interests, providing a clear view of their inherent preferences and enhancing user trust in the system.

6.4 Experiments

To validate the effectiveness of our proposed CAST-Rec, we have conducted extensive experiments on two real-world news recommendation datasets to answer the following research questions.

- **RQ1**: How does our proposed CAST-Rec perform in the news recommendation task compared to state-of-the-art (SOTA) news recommenders?
- **RQ2**: How do different large language models (LLMs) impact the performance of CAST-Rec?
- **RQ3**: How do different hyperparameters in CAST-Rec affect its performance?
- **RQ4**: How efficient is CAST-Rec?

6.4.1 Experimental Setups

This section presents the datasets, baselines, evaluation protocol, and implementation details used in our experiments and comparisons to validate the effectiveness of CAST-Rec.

6.4.1.1 Datasets

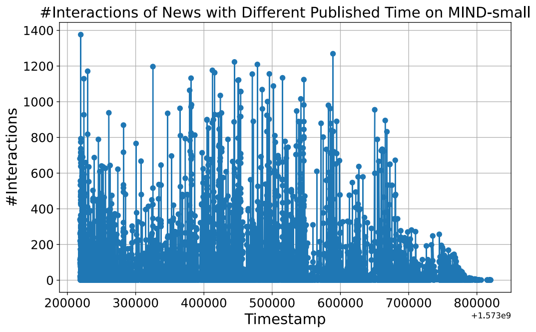
We evaluate the effectiveness of our proposed CAST-Rec on the two public news recommendation datasets: MIND-small and MIND-large [156]. MIND dataset is collected from user behavior logs on the Microsoft News platform. The author randomly sampled 1 million active users with more than five news clicks on the platform during the period from October 12 to November 22, 2019. Similarly, the MIND-small version randomly samples 50,000 active users and their behavior logs. For each user, click behaviors before this time period are regarded as history, while exposed news and user interactions in this period are formatted into impressions. Different from the original MIND dataset, we organize the impressions into a series of accumulated sequences arranged chronologically by the timestamp of each impression. To exclude very short impression sequences [5, 109], we filtered out those with fewer than 10 behavior logs. Following common practices in the sequential recommendation, we compile the most recently clicked news for each user into the test set and the second most recently clicked news into the validation set.

Furthermore, MIND contains over 160,000 English news articles, each featuring rich textual content, including the title, abstract, and category. The popularity of each news article is calculated based on its click-through rate within a recent time window [25, 106]. The recency of each news article is represented by the time elapsed since its initial exposure in the system, regardless of whether it was clicked [84, 106]. Details of MIND-small and MIND-large datasets used in our experiments are formatted in Table 6.1. The interaction distributions of news in these two datasets are visualized in Figure 6.5. It can be observed that there is a complex correlation between the publish time of news items and the corresponding interactions, indicating that relying solely on recency as the time-variant covariate is insufficient for modeling time-aware causal influences. Additionally, compared to MIND-small, users' interactions in the MIND-large dataset have higher variance.

Table 6.1: Statistical information of datasets.

Datasets	#Users	#Items	#Interactions	Density
MIND-small	1809	12668	37801	1.80×10^{-4}
MIND-large	110780	26313	1305190	0.69×10^{-4}

[User-item interaction distribution of news in `MIND-small`.]



[User-item interaction distribution of news in `MIND-large`.]

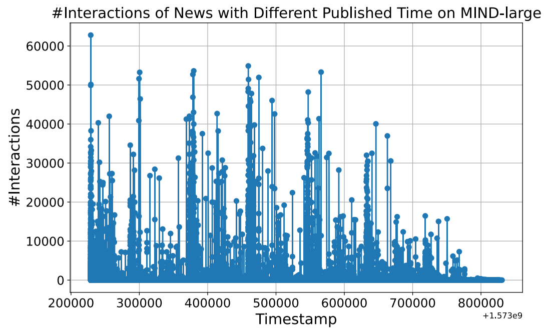


Figure 6.5: Interaction distributions with respect to news items in the two datasets.

6.4.1.2 Baselines

In our experiments, we have compared our proposed CAST-Rec with three categories of News Recommendation Systems (NRSs) as follows.

Traditional News Recommendation Systems represent user preferences based on news textual information while ignoring the evolving time. We selected four common news recommenders:

- DKN [140] utilizes a knowledge-aware convolutional neural network that integrates semantics and knowledge-level representations of news and an attention module to dynamically aggregate use history with current candidate news.
- NPA [154] integrates a CNN-based news representation model and a user representation model with word- and news-level attention mechanisms, enhanced by a personalized attention network using user ID embeddings to tailor the representation of news and users.
- NRMS [155] leverages multi-head self-attention in encoding the news titles and user browsing history, supplemented by additive attention to highlight important words in news content for more informative representations.
- NAML [153] unifies the representations of news titles, bodies, and categories from user browsing history with attention at both word- and user-interest levels.

Sequential News Recommendation Systems incorporate sequential information from user browsing history in addition to news content. We include three models in this category:

- LSTUR [5] learns long-term user representations from user ID embeddings and encodes short-term user representations from history-browsed news contents.
- CAUM [107] incorporates a candidate-aware CNN network for modeling short-term user interests and a candidate-aware attention network for aggregating previously clicked news to build a comprehensive candidate-aware user representation.
- MINS [143] designs a GRU-based interest network to extract potential multiple interests of users from their historical browsed sequences.

Temporal News Recommendation Systems consider temporal data into predictions, represented by the following two methods:

- PP-Rec [106] combines a personalized matching score based on user browsing history with a news popularity score that encodes click-through rates, recency, and news content.
- TCCM [25] analyzes the causal effects of time, news popularity, and the alignment score between news content and user interest in user behaviors, and it also estimates news popularity by learning from the granularity of entities and words.

6.4.1.3 Evaluation Protocol

The recommendation performance of our CAST-Rec is evaluated using the following metrics, all of which are commonly employed in similar tasks in [25, 74, 106, 154].

- Mean Reciprocal Rank (MRR) considers the rank position of the first target item in the ranked list. A higher MRR score indicates better performance.
- Area Under the Curve (AUC) takes both the true positive rate and false positive rate into account. A model with a higher AUC is better at distinguishing between positively interacted items and negative items.
- Normalized Discounted Cumulative Gain (NDCG)@K accounts for the position of target items in the top-K ranked recommendation list and assigns higher importance to items in the front positions. Following previous works, we consider both NDCG@5 and NDCG@10, where a higher score represents better ranking performance.

6.4.1.4 Implementation Details

Our proposed CAST-Rec model is implemented using PyTorch and is trained on a single NVIDIA GeForce RTX 3090 GPU. The learning rate for CAST-Rec is selected from the values [5e-3, 3e-3, 1e-3, 5e-4, 3e-4, 1e-4], while the batch size is chosen from [64, 128, 256], and the dimensionality is selected from [8, 16, 32]. The model employs a dropout rate of 0.1, with a maximum sequence length of 5. We utilize 2 heads for the multi-head attention layers in CAST-Rec. For a fair comparison, all other baseline models are also trained on a single NVIDIA GeForce RTX 3090 GPU. Additionally, we adopts LLaMA-2-7B, LLaMA-2-13B, Vicuna-13B models from Hugging Face for encoding time-invariant information. Specifically, the result as detailed in Table 6.2 is based on the inference on Vicuna-13B. The large language model inference is conducted using one NVIDIA A800 GPU. This setup ensures that the performance of CAST-Rec and the baseline models is evaluated under comparable hardware conditions, allowing for a rigorous assessment of the proposed model's capabilities.

6.4.2 Recommendation Performance (RQ1)

Table 6.2: Recommendation Performance Comparison of CAST-Rec with 9 baseline models on `MIND-small` and `MIND-large` datasets. Bold font is used to denote the best performance, while underlined font represents the second-best performance.

Model		**MIND-small**				**MIND-large**			
		MRR	AUC	NDCG@5	NDCG@10	MRR	AUC	NDCG@5	NDCG@10
Traditional NRS	DKN	0.3092	0.6011	0.3312	0.3936	0.3217	0.6196	0.3479	0.4055
	NPA	0.3163	0.6329	0.3522	0.4001	0.3309	0.6542	0.3671	0.4197
	NRMS	0.3122	0.6407	0.3457	0.4066	0.3310	0.6602	0.3599	0.4201
	NAML	0.3451	0.6631	0.3699	0.4309	0.3620	0.6834	0.3823	0.4652
Sequential NRS	LSTUR	0.3466	0.6810	0.3812	0.4428	0.3612	0.6982	0.3901	0.4519
	CAUM	0.3513	0.6819	0.3892	0.4557	0.3723	0.7059	0.4099	0.4698
	MINS	0.3562	0.6863	0.3907	0.4598	0.3707	0.7068	0.4102	0.4744
Temporal NRS	PP-Rec	0.3909	0.7112	0.4298	0.4966	0.4179	0.7302	0.4470	0.5098
	TCCM	<u>0.4198</u>	<u>0.7235</u>	<u>0.4651</u>	<u>0.5291</u>	<u>0.4402</u>	<u>0.7499</u>	<u>0.4896</u>	<u>0.5467</u>
Ours	CAST-Rec	**0.4513**	**0.7303**	**0.4992**	**0.5419**	**0.4697**	**0.7521**	**0.5186**	**0.5609**
	Improv.	7.50%	0.94%	7.33%	2.42%	6.70%	0.29%	5.92%	2.60%

In this section, we evaluate the recommendation performance of CAST-Rec by comparing it with nine state-of-the-art (SOTA) news recommendation systems. These baseline models include a range of approaches, from traditional news content-matching models to sequential and temporal news recommendation models. We conducted experiments using widely recognized real-world news recommendation datasets, MIND-small and

MIND-large, to ensure a robust evaluation. We compare the models based on three key metrics: Mean Reciprocal Rank (MRR), Area Under the Curve (AUC), and Top-K Normalized Discounted Cumulative Gain (NDCG@K). Consistent with previous studies [25, 106, 154], we set K=[5, 10] for our evaluations. The comprehensive comparison aims to highlight how CAST-Rec performs relative to existing models in capturing user preferences and delivering relevant news recommendations. Through experiments on MIND-small and MIND-large datasets, we provide insights into the effectiveness of our model across different dataset sizes.

We present the experimental results in Table 6.2. When compared to the best performance of previous models, specifically TCCM [25], our proposed CAST-Rec achieves notable improvements. Specifically, CAST-Rec outperforms TCCM by 7.50% and 6.70% in MRR and by 7.33% and 5.92% in NDCG@5 on the MIND-small and MIND-large datasets, respectively. These advancements can be attributed to two primary factors:

- **Modeling Evolving Causal Dependencies**: CAST-Rec effectively captures the evolving causal relationships between exposed news content and two time-variant covariates, thereby enhancing the modeling of user preferences over time. We achieve this by modeling both intra- and inter-dependencies among exposed news, news popularity, and news recency using masked self-attention and cross-attention layers within multi-input causal blocks.

- **Leveraging Large Language Models**: We utilize the semantic understanding capabilities of LLMs to generate concise summaries that encapsulate users' essential interests, which are then continuously refined over time. By predicting these distilled interests, the system is more likely to select the top-K news items that satisfy user preferences. The contribution and detailed analysis of using LLMs are discussed in Section 6.4.3.

Despite the significant improvements in MRR and NDCG@5, the enhancements in AUC and NDCG@10 are relatively modest, limited to 0.94% and 0.29% in AUC, 2.42% and 2.60% in NDCG@10 on the two datasets, respectively. This indicates a reduced improvement in the overall ranking of all candidate news items compared to the enhancement seen for the news items ranked at the top. We attribute this to the following two factors: Firstly, CAST-Rec emphasizes items in the front positions due to a relatively small sequence length (5 in experiments). Secondly, the baselines already demonstrate excellent overall ranking performance, making further improvements challenging. In addition, compared to MIND-small, the improvements on MIND-large are not as significant. A possible reason for this diminished improvement is that the MIND-large dataset contains more inactive users, resulting in a more long-tailed distribution, as evidenced in Table 6.1 and Figure 6.5.

Table 6.2 also provides insights into the impact of incorporating temporal information in news recommendations. Traditional NRSs (DKN [140], NPA [154], NRMS [155], NAML [153]) rely on personalized attention mechanisms to predict user preferences. However, they face performance bottlenecks. As the best model in the first category, NAML [153] achieves around 0.37 in the MIND-small dataset and 0.38 in the MIND-large dataset, with regard to NDCG@5. The second category of NRSs, including LSTUR [5], CAUM [107], and MINS [143], introduces sequential patterns in user interactions. These models capture the order of user actions but do not fully account for the time-variant factors related to the news items. For example, MINS [143] enhances recommendation performance by 5.62% in MIND-small and 7.30% in MIND-large datasets with regard to NDCG@5 by extracting multiple user interests using several GRUs from their browsing history. Temporal NRSs, such as PP-Rec [106] and TCCM [25], achieve the highest performance by modeling both the sequential and temporal aspects of evolving user preferences. These models consider not only the order of interactions but also the changing nature of user interests and news popularity over time. Specifically, PP-Rec [106] makes predictions by combining popularity scores with personalized matching scores, while TCCM [25] further advances this approach with causal interventions, leading to significant performance improvements. TCCM [25] achieves the best performance among baselines with an NDCG@5 score of 0.4651 in the MIND-small dataset, respectively. These insights underscore the importance of integrating temporal information into news recommendation systems to better capture the evolving nature of user interest and news content.

6.4.3 Different Large Language Models (RQ2)

In this section, we delve into the performance of our proposed CAST-Rec framework when enhanced with different Large Language Models, conducting both ablation and case studies for analysis. Using metrics such as MRR, AUC, NDCG@5, and NDCG@10, we quantitatively assess recommendation performance under the enhancements of different LLMs. Additionally, we conduct a case study to closely examine the details in the summaries generated by these LLMs.

Table 6.3: Ablation studies comparing the recommendation performance of random user interest initialization versus initialization using summaries generated by different LLMs within our proposed CAST-Rec framework, evaluated on the `MIND-small` dataset.

Models	MRR	AUC	NDCG@5	NDCG@10
random	0.4012	0.7109	0.4315	0.4661
LLaMA-2-7B	0.4295 +7.05%	0.7149 +0.56%	0.4619 +7.05%	0.5007 +7.42%
LLaMA-3-8B	0.4313 +7.50%	0.7183 +1.04%	0.4622 +7.11%	0.5019 +7.68%
LLaMA-2-13B	0.4487 +11.84%	0.7201 +1.29%	0.4893 +13.40%	0.5272 +13.11%
Vicuna-13B	0.4513 +12.49%	0.7303 +2.73%	0.4992 +15.69%	0.5419 +16.26%

6.4.3.1 Ablation Study

We begin with an ablation study on several LLMs of different scales, demonstrating the effectiveness of introducing LLMs in our CAST-Rec. In particular, we consider four different models, i.e., LLaMA-2-7B [130], LLaMA-3-8B [90], LLaMA-2-13B [130], Vicuna-13B [137].

- LLaMA-2-7B [130] is an autoregressive model in the LLaMA series [128] that utilizes large-scale transformer architecture. LLaMA-2 [130] performs well in various natural language tasks through supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to align generated texts with user instructions. LLaMA-2-7B consists of 7 billion parameters, balancing computational costs and semantics processing capability.

- LLaMA-3-8B [90] is an autoregressive model trained with 8 billion parameters, building on its predecessors in the LLaMA family. It is also fine-tuned using SFT and RLHF, featuring an expanded context window and enhanced reasoning capabilities, making it particularly useful for complex natural language processing tasks.

- Vicuna-13B [137] is fine-tuned on 13 billion parameters using using 125,000 user-shared conversational data. This enables it to achieve impressive performance in various natural language tasks, such as question answering.

- LLaMA-2-13B [130] presents superior performance in various natural language tasks compared to LLaMA-2-7B, owing to its larger parameter scale of 13 billion.

We replace the method for summarizing time-invariant user interests from browsing history with the aforementioned four LLMs, alongside the performance of random user interest initialization (without summarization, denoted as "random" in Table 6.3). The performance of CAST-Rec using these five different methods to capture time-invariant user interests is evaluated using MRR, AUC, NDCG@5, and NDCG@10, as shown in Table 6.3. Due to the significant computational costs, we limited our experiments to the MIND-small dataset. Our findings are as follows.

6.4.3.2 Case Study

To control the quality of the time-invariant covariates extracted by LLMs, in addition to using the hidden states, we also examine the corresponding generated texts. Specifically, we randomly select two users, `user 21` and `user 883`, from the MIND-small dataset and compare the texts generated by different LLMs.

In Figure 6.6, the red squares represent the news titles arranged in the order of the user's browsing history, while the blue squares include the texts generated by LLMs. Content within the same news topic is highlighted using the same text color, while incorrect topics generated by the LLMs are specifically highlighted in gray. `User 21` appears to prefer news related to social issues (red), celebrities (green), and events in the US, especially in the Bay Area (blue). In contrast, `user 883` shows a broader range of interests, favoring news about politics (red), entertainment (green), sports (blue), and lifestyles (purple). It is observed that LLMs with smaller parameter sizes (LLaMA-2-7B and LLaMA-3-8B) are more prone to generating incorrect topics, leading to poorer performance. In contrast, larger LLMs can effectively capture users' time-invariant preferences through concise sentences, thereby improving user modeling in news recommendations. Moreover, LLMs are adept at summarizing various entities within a single topic across different news titles using general terms, which helps address the challenge of managing diverse textual representations in recommendation systems. Additionally, these LLM-generated summaries are more informative than the annotated categories of each news article.

6.4.4 Hyperparameter Study (RQ3)

In this section, we have conducted studies on two hyperparameters to investigate their influence on recommendation performances: the dimension of hidden representations and the max length of user historical browsing sequences.

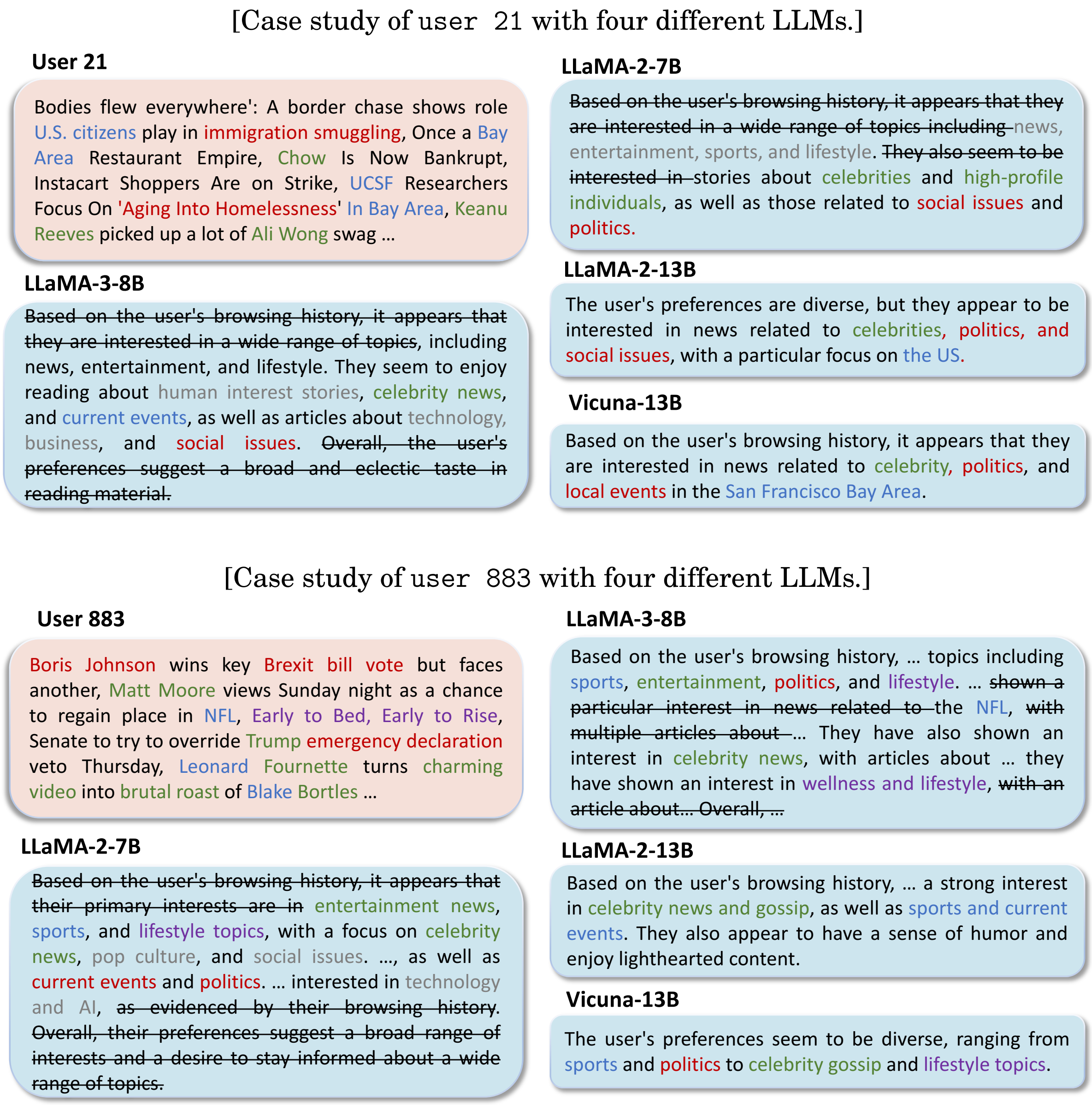


Figure 6.6: Case studies of the time-invariant features extracted by different LLMs for users.

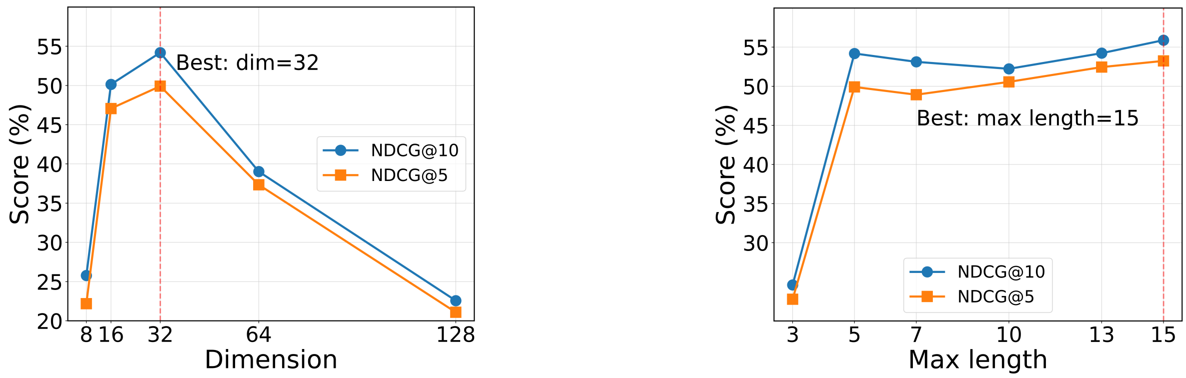


Figure 6.7: The influence of different values of hyperparameters `dimension` (left) and `the maximum sequence length` (right) on recommendation performances evaluated by NDCG@5 and NDCG@10 on MIND-small.

6.4.4.1 Different Dimensions Influence on Recommendation

The left plot in Figure 6.7 illustrates the effect of hidden representation dimensions on recommendation performance, measured by NDCG scores at the top-5 and top-10 lists (NDCG@5, NDCG@10). A dimension of 32 yields the best performance. This might be because a lower dimension (8) lacks the capacity to capture the complex dependencies between variables, while a higher dimension (128) suffers from data sparsity or difficulty in tuning other hyperparameters. The dimension is set to 32 in our experiments.

6.4.4.2 Different Sequence Lengths Influence on Recommendation

When modeling with multi-input causal blocks, we first divide the user browsing history into a series of subsequences. The right plot in Figure 6.7 shows the effects of the maximum length of these subsequences on recommendation performance, measured by NDCG@5 and NDCG@10. Short sequences fragment the sequential information in users' click behaviors, while longer sequences tend to be more expressive. Performance plateaus when the maximum length is set to 5 and peaks at a length of 15. However, due to time and memory limitations, we chose a maximum length of 5 in our experiments.

6.4.5 Efficiency Analysis (RQ4)

Time Complexity. The time complexity of CAST-Rec comes from the following processes: 1) time-aware modeling with L number of multi-input causal blocks (Section 6.3.3) 2) recommendations on $|\mathcal{V}|$ number of candidate news items (Section 6.3.4). Here, we use T to represent the length of user's browsing history, d to represent the hidden dimension

Component	**Parameter Sizes**
LLM (depends on the model adopted)	8B
Causal transformer blocks (×3)	0.02MB
Recommendation layer	2.39MB
User and item embeddings	119.04MB

Table 6.4: Parameter sizes of the different components in CAST-Rec using LLaMA-3-8B.

in each respective model. For the first process, the time complexity in each causal block is mainly contributing to the multi-head self-/cross-attention layers, with $O(T^2Nd)$ each and $O(3T^2Nd)$ in total, where N denotes the number of heads. Subsequently, the feed-forward layer consumes $O(Td^2)$. The L numbers of causal blocks with three inputs inside each multiply these values, yielding $O(3L(3T^2Nd + Td^2))$. For the second process, the matrix factorization in user satisfaction score calculating costs $O(2|\mathcal{V}|d)$. In total, the time complexity for CAST-Rec is $O(2|\mathcal{V}|d + 3L(3T^2Nd + Td^2))$. Given the number of heads is set as 2 ($N = 2$) in our proposed CAST-Rec, and the number of multi-input causal blocks is set as 2 ($L = 2$), the time complexity can be simplified as $O(|\mathcal{V}|d + T^2d + Td^2)$.

Computational Efficiency. We provide the parameter sizes of different components in Table 6.4, showing that the primary computational and memory costs stem from the adoption of large language models (LLMs). In CAST-Rec, both the computational and time overhead can be mitigated by asynchronously utilizing APIs for LLMs. We conducted our experiments by deploying open-source LLMs locally; however, our model is designed to accommodate API usage in scenarios with limited time and computational resources.

CHAPTER 7

Conclusion and Future Works

This chapter concludes our collective efforts at the data and representation levels to enhance structured knowledge reasoning in deep models. It also outlines forward-looking directions at the model level that can further expand the boundaries of structured reasoning and foster interdisciplinary applications of this work.

7.1 Conclusion

This thesis has presented novel frameworks and methodologies for representing and reasoning over structured knowledge, with particular emphasis on improving the interpretability, adaptability, and robustness of recommender systems and large language models (LLMs). The research contributes to both data-level and representation-level understanding of structured reasoning, bridging the gap between structured data representations and the reasoning capabilities of modern deep models.

At the data level, the introduction of StructFact, a benchmark specifically designed for evaluating LLMs' reasoning over structured data, offers a rigorous foundation for understanding how models process heterogeneous structures, topological dependencies, order invariance, and sparsity. Beyond technical insights, StructFact provides a diagnostic tool for evaluating LLMs in high-stakes domains such as healthcare, finance, and scientific discovery, where factual precision and structural understanding are vital. These findings have implications that extend beyond model evaluation. StructFact identifies both strengths and limitations in real-world applications, such as healthcare and finance, where precise factual reasoning is crucial.

At the representation level, this thesis advances several approaches to improve deep models' ability to reason over structured and dynamic knowledge. The C-MBR model introduces causal reasoning mechanisms to mitigate confounding effects from heterogeneous user behaviors in multi-behavior recommendation. The HyperG framework models complex structural relationships through hypergraph generation, augmenting LLMs with the capacity to reason over sparse, hierarchical, and interconnected structures. Meanwhile, CAST-Rec captures temporal and contextual evolutions in user behavior by unifying representations from LLMs and transformers. Collectively, these methods not only improve structured reasoning performance but also lay the groundwork for cross-domain generalization, enabling structured reasoning to be applied across diverse fields—from recommendation to time-series forecasting, knowledge graph reasoning, and beyond.

In summary, this thesis offers a unified perspective on how structured knowledge can be effectively represented, reasoned over, utilized in deep models. By developing benchmarks, algorithms, and models that bridge data and representation levels, this work provides a foundation for building more interpretable, data-efficient, and generalizable deep models. The broader vision is to enable AI systems that can reason about structured phenomena in the real world, supporting applications in domains such as medicine, social science, and computational biology, where complex relational structures and causal reasoning are essential.

7.2 Future Works

In the future, we will extend our research on structured knowledge to the model level. Future studies in model-level advancements will focus on leveraging the collaboration between deep models to enhance the representation of structured knowledge and improve decision-making for downstream tasks. By integrating multiple deep learning models, such as combining multiple LLMs as the agent, we can capture deeper and multi-faced understanding of structured data. This collaborative approach will facilitate the extraction of meaningful dependencies, interrelationships, and contextual knowledge from diverse data sources, enabling models to make more informed and accurate decisions.

Moreover, this model-level perspective opens pathways for cross-disciplinary innovation. For instance, integrating structured reasoning with domains like computational social science, biomedical informatics, and medical science could allow these deep-learning-based AI systems to analyze multi-modal, relational, and context-dependent data with unprecedented depth. Coupling structured reasoning with causal discovery and counterfactual analysis will also support the development of scientifically grounded AI, capable of explaining and generalizing across real-world phenomena.

Looking forward, this research paves the way for a new generation of structure-aware and reasoning-centric AI systems that are capable not only of prediction and recommendation but also of understanding, explanation, and adaptation within the structured complexity of human knowledge and real-world environments.

APPENDIX A

SUPPLEMENTARY MATERIALS FOR STRUCTFACT

A.1 Implementation Details

We use 32GB memory with Ubuntu 20.04 LTS (a open-source Operating System using the Linux kernel and based on Debian) and 4 Nvidia A800 with 80GB memory for inference. we adopt vllm [67] 0.5.4 to speed up inference. All models share a set of hyperparameters, as detailed in Table A.1.

Hyperparameter	**Value**
`top_p`	0.95
`temperature`	0.6
`max_generation_token (w/o CoT)`	10
`max_generation_token (w/ CoT)`	512
`max_evidence_token`	2500

Table A.1: Hyperparameters of LLMs

A.2 Future Directions

Apart from our findings, we propose future directions to advance LLM use in knowledge-sensitive tasks involving structured data.

The performance of LLMs diminishes as the availability of structured evidence shifts from abundant to absent. Given the limited improvements achieved through prompt engineering on instruction-tuned models, incorporating an additional structure-aware module may offer a more effective approach for learning from structured data. Such specialized modules enable task-adaptive learning and knowledge transfer while keeping computational and time costs manageable. Furthermore, LLMs show considerable potential in leveraging unstructured knowledge to complement structured data. A key challenge in this process is avoiding the distortion of precise knowledge contained within the structured data. Future research could explore the use of reinforcement learning to iteratively correct distortions in reasoning.

A.3 Generation Randomness

To assess the randomness of generation, we conducted zero-temperature experiments on the 10 LLMs and reported the results in Table A.2.

Methods	Zero-shot w/o CoT		Zero-shot w/ CoT		Few-shot w/o CoT		Few-shot w/ CoT		Overall	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Qwen2-7B	38.12	48.51	45.21	48.34	52.15	55.89	50.18	53.97	46.41	51.68
LLaMA-3-8B	27.89	32.01	25.98	33.82	28.76	35.04	47.33	48.71	32.49	37.40
Gemma-2-9B	31.45	34.77	42.06	47.12	43.15	47.92	53.14	54.32	42.45	46.03
Qwen2-7B Instruct	46.30	52.15	43.01	50.26	44.67	50.83	42.95	51.18	44.23	51.11
LLaMA-3-8B Instruct	60.88	58.55	42.05	49.73	60.32	58.97	49.12	57.32	53.09	56.14
Gemma-2-9B It	43.01	49.27	47.21	56.05	45.12	53.06	49.32	58.01	46.17	54.10
GLM-4-9B Chat	56.85	59.57	47.93	56.84	54.45	57.68	51.84	59.43	52.77	58.38
Mistral-7B Instruct	48.01	50.12	39.87	49.12	55.15	56.82	46.35	54.78	47.34	52.71
GPT-3.5-turbo	59.92	60.96	59.02	63.11	52.37	57.00	63.01	66.22	58.58	61.82
GPT-4o-mini	65.32	67.41	62.15	67.58	61.89	65.32	64.48	69.12	63.46	67.36

Table A.2: Performance of 10 LLMs on using various prompts (temperature=0).

A.4 Evaluation Protocol

In this paper, we use six different metrics for evaluating the reasoning performance of LLMs on structured knowledge. We formulate all the evaluation metrics used in this section.

- Accuracy.

  (A.1)
  $$Acc. = \frac{TP + TN}{TP + TN + FP + FN}$$
  where TP, TN, FP, FN represent the number of true positive, true negative, false positive, and false negative, respectively.

- Weighted F1 score.

  (A.2) $$F1 = \sum_{i=1}^{N} \frac{n_i}{N} F1_i$$

  where n_i is the number of samples in label i, N is the number of all samples, $F1_i$ is the F1 score for label i.

- Balanced accuracy.

  (A.3) $$BA = \frac{1}{N} \sum_{i=1}^{N} (TPR_i), TPR = \frac{TP}{TP+FN}$$

  where TPR_i is the true positive rate of label i.

- Macro F1 score.

  (A.4) $$MacroF1 = \frac{1}{N} \sum_{i=1}^{N} F1_i$$

- Precision.

  (A.5) $$Prec. = \frac{TP}{TP+FP}$$

- Recall.

  (A.6) $$Recall = \frac{TP}{TP+FN}$$

A.5 Detailed Introduction to selected LLMs

Meta's Llama series, including Llama 2 and Llama 3 [129], released in 2023 and 2024, are designed for various tasks like text generation and programming. Llama3 is designed to be more intelligent, faster, and more versatile, making it suitable for a wide range of applications. Qwen2 [163] [**?**] is a strong language models developed by Alibaba Cloud, showing state-of-the-art performance in several benchmarks, especially in coding and mathematics. ChatGLM3 [38] is the latest generation of pre-trained dialogue models developed by Zhipu AI in collaboration with Tsinghua University's Knowledge Engineering Group (KEG). Developed by OpenAI, GPT-4o-mini [98] is its most cost-efficient small model in the GPT series, featuring enhanced context understanding and text generation capabilities, scoring 82% on MMLU [44]. Gemma2 [125] is Google's latest iteration of open large language models (LLMs), building on the success of the original Gemma series. Coming with two sizes, 9 billion and 27 billion parameters, each size has a base model (pre-trained) and an instruction-tuned version.

Model Name	**Company**	**Cut-off Date**	**Release Date**
Qwen2-7B	Alibaba	2023	February 2024
LLaMA-3-8B	Meta	March 2023	April 2024
Gemma-2-9B	Meta	-	July 2024
Qwen2-7B-Instruct	Alibaba	2023	February 2024
LLaMA-3-8B Instruct	Meta	March 2023	Apirl 2024
Gemma-2-9B It	Meta	-	July 2024
GLM-4-9B Chat	Zhipu AI	-	June 2024
Mistral-7B Instruct	Mistral AI	-	September 2023
GPT-3.5 Turbo	OpenAI	September 2021	November 2022
GPT-4o mini	OpenAI	October 2023	May 2024

Table A.3: Table of the selected LLMs, companies, knowledge cut-off dates, and released dates.

A.6 Ethical Statement

We affirm that our StructFact benchmark is constructed using open-source datasets and adheres to the CC-BY-4.0 license. To uphold privacy and confidentiality, we have ensured that our dataset contains no direct or indirect sensitive personal information. Users accessing our StructFact should ensure that no personally identifiable information or toxic content is included.

Our research postulate that our StructFact benchmark is under an environment devoid of possible attacks. However, given that the structured data in our proposed benchmark is sourced from publicly editable WikiPedia pages, it is inherently vulnerable to various threats, including adversarial attacks. Intended attacks, such as data poisoning, involve malicious actors deliberately inserting false or misleading information or altering existing structured data. These actions can compromise the integrity of the data, distorting the knowledge within LLMs and undermining accurate factual reasoning. Unintentional attacks, such as accidental data deletion or incorrect data entry, also pose significant risks. These errors can degrade both the quality and structure of the data, potentially leading LLMs to draw incorrect inferences, thus might compromising the overall factuality of the benchmark.

Moreover, while the questions in our StructFact benchmark reflect real-world facts, they do not originate from practical applications. Therefore, we offer StructFact as a resource to guide users in their inferences, without claiming to provide absolute assertions. We advise against using StructFact as a basis for developing models intended to verify facts in real-world applications.

A.7 Task Categorization for Fine-Grained Analysis

To perform a fine-grained analysis of LLMs' reasoning capabilities over structured data, we refined each major task into specific subcategories. This section outlines the methodologies employed for task refinement.

A.7.1 Arithmetic Calculation

Following [22, 45], we refined the *Arithmetic Calculation* task into three distinct subcategories based on the complexity and nature of mathematical reasoning required:

1. **Numerical Matching**: Tasks involving the direct matching and verification of a specific value within the data, typically a precise, singular number, without any calculations or statistical analyses.
    - *Example*: "Is the number of gold medals won by China in the 2024 Olympics 22?"
2. **Numerical Comparison**: Tasks requiring the comparison of numerical values to determine size, quantity, or order, including comparisons involving superlatives.
    - *Example*: "Did China win more gold medals than any other country in the 2024 Olympics?"
3. **Computational Analysis**: Tasks necessitating calculations, statistical methods, or logical reasoning to perform aggregations, averages, percentage calculations, or other mathematical analyses. This includes basic arithmetic operations (addition, subtraction, multiplication, and division) and conditional judgments.
    - *Example 1*: "Does the percentage of gold medals won by China exceed 10% of the total in the 2024 Olympics?"
    - *Example 2*: "Did China win 4 more gold medals than France in the 2024 Olympics"

To annotate the dataset according to these refined categories, we established precise definitions and guidelines for each subcategory. We then employed GPT-4o for automatic labeling of the tasks, followed by independent reviews from three experts. Discrepancies identified by the majority of experts were manually corrected. This combination of automated labeling and expert validation ensured both efficiency and high accuracy, aligning with best practices in semi-automated annotation workflows.

A.7.2 Geography-time Reasoning

We focused on identifying and categorizing named entities related to temporal and spatial information within the questions. Utilizing the Named Entity Recognition (NER) tool from SpaCy, specifically the `en_core_web_sm` model[1], we automatically recognized entities in the text. Based on the NER tags assigned by SpaCy, we refined the questions into:

- **Temporal Entities**: Questions containing temporal entities such as dates (`DATE`) and times (`TIME`).
- **Spatial Entities**: Questions involving spatial entities, including geopolitical entities (`GPE`), natural locations like mountains and rivers (`LOC`), and facilities or artificial landmarks (`FAC`).
- **Spatiotemporal Entities**: Questions that contain both temporal and spatial entities, such as combinations of `DATE+GPE`, `DATE+LOC`, or `DATE+FAC`.

A.7.3 Multi-hop Reasoning

Multi-hop Reasoning refers to the process of linking and integrating information from multiple sources or steps to arrive at a final answer [47, 133, 168]. It requires the system to go beyond simple, one-step reasoning (single-hop) by making logical connections across various pieces of information that are distributed across multiple documents, sentences, or structured data points.

In our work, we refined the *Multi-hop Reasoning* task by categorizing questions based on the number of reasoning steps, or "hops," required to derive the correct answer. Each "hop" is defined as a step where the model must integrate information from two distinct data sources within the structured data, such as table cells, headers, captions, or list items. Using a rule-based method, we classified the questions into six categories: 1-hop, 2-hop, 3-hop, 4-hop, 5-hop, and greater than 5 hops. This systematic classification

[1] `https://spacy.io/usage/models`

allowed us to assess the complexity of the reasoning required and analyze the model's capacity to perform multi-hop reasoning over structured data.

A.7.4 Composition Understanding

Inspired by [117], We examined the model's ability to reason over structured data compositions with irregularities. Specific categorization rules were defined to identify different types of compositional challenges:

1. **Complex Structures**: Compositions involving intricate dependencies, such as a single table cell spanning multiple columns or rows, nested tables, or hierarchical lists.

2. **Missing Values**: Instances where the structured data contains unknown or missing values, requiring the model to handle incomplete data.

3. **Incomplete Descriptions**: Cases where the structured data includes ambiguous or insufficient descriptions, such as unclear column headers or labels.

Two experts manually annotated the dataset according to these categories and conducted a cross-review to ensure consistency and resolve discrepancies. This rigorous annotation process enhanced the reliability of our categorization.

A.7.5 Combining Structured and Unstructured Data

Given the wide existence of semi-structured data [23, 94, 100], the *Combining Structured and Unstructured Data* task aimed to assess the model's ability to integrate information from both structured data (e.g., tables) and accompanying unstructured textual context. We designed three experimental conditions:

- **Original Unstructured Context**: Using the original unstructured context provided in the dataset alongside the structured data.

- **Enhanced Unstructured Context**: Augmenting the unstructured context by generating additional descriptions of the structured data using GPT-4o, including details such as table formats and relevant contextual information to provide extra background knowledge.

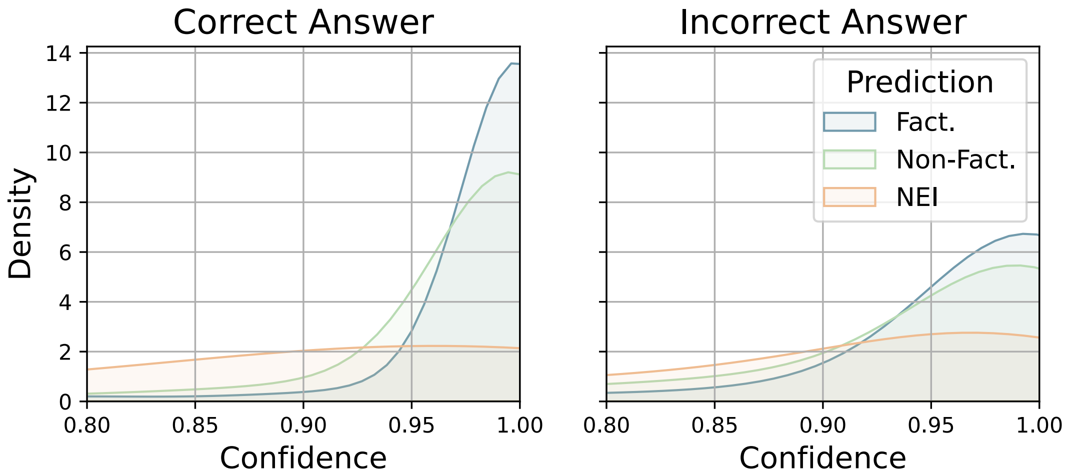


Figure A.1: Confidence distribution of answers.

- **No Unstructured Context**: Removing all unstructured context, leaving only the structured data for the model to process.

By varying the availability and richness of unstructured context, we aimed to analyze how different levels of contextual information affect the model's ability to reason over structured data.

A.8 Model Confidence

Towards reliable reasoning outcomes, we also concern about the confidence of LLMs' outputs. Specifically, We used the probability of the model's first token output corresponding to the predicted label as the confidence score. Figure A.1 illustrates that the model shows higher confidence with its predicted facts, compared to those are predicted as non-factual. For NEI responses, whether predicted correctly or not, the confidence remains low with only minor variations. Notably, the models exhibit uncertainty in their incorrect answers, as evidenced by a significant drop in confidence, especially for factual and non-factual predictions. This indicates that the model's confidence levels are somehow aligned with the accuracy of their predictions, which could be useful for gauging the reliability of the model's answer.

A.9 Case Study

Please see figs. A.2 to A.6 for case studies for each task and the responses from different LLMs.

A.10 Prompt Strategies Analysis

A.10.1 Detailed Introduction to Employed Prompts

Each LLM in our main result decipted in Table 6.2 is experimented with different prompting strategies: Zero-shot without CoT [62], Zero-shot with CoT, Few-shot with CoT, Few-shot with CoT. All the strategies used in this paper begin with an instruction denoted as p = "*You will be given with a question. Please response with 'Yes', 'No', or 'Not Sure Enough'.*" For any input question $q_i \in \mathcal{Q}$, structural data $d_i \in \mathcal{D}$ the model $LLM(\cdot)$ is expected to generate an answer $y_i \in \mathcal{Y} = \{$`'Yes'`,`'No'`,`'Not Sure Enough'`$\}$. Each question is categorized into one task t from the five aforementioned reasoning tasks in $\mathcal{T}$. Examples of the prompts used in our experiments are shown in Figure A.8.

A.10.1.1 Prompts in Main Results

Prompt with Zero-shot. In the prompting strategy with zero-shot setting, the LLM is expected to output the answer y_i to the question q_i directly, formally, $y_i = LLM(p, q_i, d_i)$. For example, the factual answer y_i = `"No"` should be responded from the LLMs when being asked with the question q_i = `"Is London the host city of the 2024 Olympic-Games?"`, together with the table of Olympic Games host cities denoted by d_i.

Prompt with Few-shot. In the few-shot prompting strategy, to guide the LLM to correctly reason, we include an example question q_x and structural data d_x together with prompt p for question q_i, where the example question q_x and question q_i fall in the same task, i.e., $q_x, q_i \in t$. This process is formulated as $y_i = LLM(p \| q_x \| d_x, q_i, d_i)$. The LLM is expected to answer with y_i = `"Yes"` when given question q_i = `"Has Paris hosted the-Olympic Games three times?"` and the table of Olympic Games host cities d_i.

Prompt with Chain of Thought (CoT). In the prompting strategy with CoT [62], a two-stage prompt is employed to derive the reasoning process along with with the answer. To guide the LLM in carefully considering the process of determining the answer y_i, the prompting sentence s = `"Let's think step by step"` is added to the question q_i, formally, $y_i = LLM(p, q_i \| s, d_i)$.

A.10.1.2 Prompts in Evidence Resilience Analysis

Prompt with Shuffled Structured Data. To investigate the performances of LLMs towards different prompting context, we shuffle the structure of data. Specifically, we shuffle the rows/columns in tables, and the elements in lists. Formally, for question q_i, the output can be presented as $y_i = LLM(p, q_i, d_i^{\prime})$, where $d_i^{\prime}$ denotes the shuffled data.

Prompt without Structured Data. Given that the structural data is sourced from Wikipedia, it is assumed that LLMs have been exposed to these data during their training phase. Therefore, we are also interested in the ability of LLMs to answer factual questions q_i without being provided with the contextual structural data d_i. The process under this strategy can be formulated as $f_5 : y_i = LLM(p, q_i)$.

Prompt with self-refinement. The self-refinement strategy is designed to enhance the performance of LLMs by prompting them to iteratively providing feedback to its previous responses. Formally, the process at n-th round of refinement can be presented as $y_i^n = LLM^n(p, q_i, d_i, r_i^{n-1})$, where r_i represents the LLM's response in the last round. In our experiments, due to constraints on computing resources and time, we set n=1.

Prompt with self-consistency. The self-consistency strategy is designed to enhance the performance of LLMs by employing majority voting on multiple rounds of queries. Assume the response from the model at the n-th round as y_n, the final prediction of LLM can be formualted as $y_{final} = argmax_{c_j} \sum_{j=1}^{k} counts(y_n = c_j)$, where c denotes the available choices of the prediction label, i.e., 'Fact.', 'Non-Fact.', and 'NEI' in this paper.

Prompt with format instructions. We also provide instructions of the formats of the structured data to the zero-shot prompts. Given format instructions as f, which illustrates how the structured data looks like, the process can be formulated as $y_i = LLM(p \| f, p_i, d_i)$.

A.10.2 Analysis towards Other Prompting Strategies

Given the successes of other CoT strategies and input data format instructions [**?**], we are interested in exploring their impact on reasoning about factual knowledge within structured data. We include three prompting strategies: (i) self-refinement [88], which guides the LLM to iteratively evaluate and refine its previous responses to reach the correct answer, (ii) self-consistency [146], which mitigates hallucination through majority voting on multiple responses from the LLM, and (iii) format instructions, which prompts with descriptions of the format of the inputted structured data. There are the following notable observations from the results in Table A.4. i) Self-consistency marginally improves performance across five tasks, with an overall enhancement of 0.23%, compared to the zero-shot results without CoT in Table 6.2. ii) Format descriptions help the LLM better interpret numerical compositions, leading to a 1.02% improvement in accuracy on Arithmetic Calculation tasks. Detailed results under these strategies please refer to Table A.4.

A.11 Supplementary Results

A.11.1 Comprehensive Results

Results for Different Prompts under Other Metrics:

Please refer to Tables A.5 and A.6.

Results for Different Tasks under Other Metrics:

Please refer to Tables A.7 to A.17.

A.11.2 Analysis of Other LLMs

Model Responses Distributions:

Please refer to Figure A.9.

Model Resilience to Evidence:

Please refer to Table A.18 and Figure A.10.

Fine-grained Studies of Different Tasks:

Please refer to Figures A.11 to A.14.

Model Confidence Analysis:

Please refer to Figure A.15.

Task: Arithmetic Calculation
Q: Is the number of employees at PGNiG SA 33,071?
A: Yes

Rank in 2011	Name of concern	Location of headquarters	Revenue (Thou. PLN)	Profit (Thou. PLN)	Employees
1.	PKN Orlen SA	Plock	79 037 121	2 396 447	4,445
2.	Lotos Group SA	Gdansk	29 258 539	584 878	5,168
3.	PGE SA	Warsaw	28 111 354	6 165 394	44,317
4.	Jeronimo Martins	Kostrzyn	25 285 407	N/A	36,419
5.	PGNiG SA	Warsaw	23 003 534	1 711 787	33,071
6.	Tauron Group SA	Katowice	20 755 222	1 565 936	26,710
7.	KGHM Polska Miedz SA	Lubin	20 097 392	13 653 597	18,578
8.	Metro Group Poland	Warsaw	17 200 000	N/A	22,556
9.	Fiat Auto Poland SA	Bielsko-Biala	16 513 651	83 919	5,303
10.	Orange Polska	Warsaw	14 922 000	1 785 000	23,805

LLaMA-3-8B
1) The question asks us to find out the number of fans that attended this game.
2) We know that there are 14 games in this season.
3) We know that the number of fans that attended the first game is 24,487.
4) We know that the number of fans that attended the second game is 48,477.
5) We know that the number of fans that attended the third game is 40,387. …
A: NEI

LLaMA-3-8B Instruct
Let's analyze the question and the given data step by step.
The question asks if the number of employees at PGNiG SA is 33,071. To answer this question, we need to find the row in the table that corresponds to PGNiG SA and check the value in the "Employees" column.
After scanning the table, we find the row with the name "PGNiG SA" in the "Name of concern" column, and the corresponding value in the "Employees" column is indeed 33,071.
A: Yes

GPT-4o-mini
To answer the question about whether the number of employees at PGNiG SA is 33,071, I will refer to the structured data provided.
In the table, the entry for PGNiG SA shows that it has 33,071 employees. This value is directly listed in the table under the "Employees" column for PGNiG SA.
Since the data clearly states that the number of employees at PGNiG SA is indeed 33,071, I can confidently answer the question.
A: Yes

Figure A.2: Case Study of Arithmetic Calculation.

Models	Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.		Overall	
		Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
GPT-4o-mini	Self-Refine.	58.03	60.73	54.44	60.08	51.61	58.81	66.91	68.17	56.51	63.37	55.36	60.38
	Self-Consis.	62.80	63.49	59.84	63.02	58.30	63.92	66.91	67.35	62.30	67.06	60.83	63.69
	Format Instruct.	62.82	63.44	58.53	61.95	57.52	63.10	66.67	67.16	61.37	66.38	60.03	63.03
LLaMA3-8B Instruct	Self-Refine.	58.05	57.74	54.87	55.40	64.71	66.75	61.20	59.46	69.83	70.94	60.23	60.44
	Self-Consis.	62.52	56.96	55.33	51.38	71.00	68.54	61.19	57.51	73.30	70.86	63.26	58.95
	Format Instruct.	61.95	55.85	51.35	46.00	68.41	66.38	60.20	54.04	71.21	68.72	60.75	55.76
Gemma-2-9B It	Self-Refine.	49.56	53.86	35.18	43.16	43.62	53.69	61.69	64.01	50.63	59.93	43.41	51.04
	Self-Consis.	51.31	53.81	33.09	39.19	44.81	54.58	60.20	62.91	52.19	60.74	43.52	49.87
	Format Instruct.	53.79	55.33	34.19	40.31	45.75	55.33	59.46	61.74	53.24	61.64	44.88	50.97
Qwen2-7B-Instruct	Self-Refine.	52.24	55.50	36.87	43.32	44.66	53.88	57.21	59.93	51.39	59.54	44.95	51.34
	Self-Consis.	55.18	57.62	40.95	47.56	47.61	56.30	57.46	61.66	53.88	61.58	48.20	54.24
	Format Instruct.	56.50	58.43	37.48	43.72	48.61	57.07	57.46	61.28	54.55	62.15	47.51	53.29

Table A.4: Performance of GPT-4o-mini using different prompting strategies across five factual tasks.

Task: Spatiotemporal Cognition
Q: Was the Khotan Mosque completed in 1870 and is it located in Khotan, Xinjiang, China?
A: Yes
Data:
* Islam in China
* List of mosques in China
	مسجد خوتان
Religion	
Affiliation	Sunni Islam
Location	
Location	China
Architecture	
Type	Mosque
Completed	1870

Qwen-2-7B
The provided data mentions that the Khotan Mosque was completed in 1870 and is located in Khotan, Xinjiang, China.
A: Yes

Qwen-2-7B Instruct
Based on the structured data provided, the Khotan Mosque is indeed located in China, and it was completed in 1870. However, the data does not specify the exact location within China as "Khotan, Xinjiang." Since Khotan is a city in Xinjiang, it can be inferred that the mosque is located in Xinjiang, China. However, the data does not explicitly state "Khotan" as part of the location.
A: NEI

GPT-4o-turbo
From the structured data provided, it is clear that the Khotan Mosque was completed in 1870 and is located in China. However, the data does not specify that the mosque is located in Khotan, Xinjiang. While it might be inferred from the name "Khotan Mosque" that it is located in Khotan, Xinjiang, the data does not explicitly confirm this location. Therefore, taking into account the explicit information given in the data, we can only confirm the completion year and the country, but not the specific location within China.
A: NEI

Figure A.3: Case Study of Geography-time Reasoning.

Methods	Zero-shot w/o CoT		Zero-shot w/ CoT		Few-shot w/o CoT		Few-shot w/ CoT		Overall	
	BA	MF1	BA	MF1	BA	MF1	BA	MF1	BA	MF1
Qwen2-7B	23.06	19.97	39.69	26.00	35.85	25.69	38.86	28.70	34.37	25.09
LLaMA-3-8B	20.65	17.57	19.17	17.94	25.29	19.66	35.16	26.76	25.07	20.48
Gemma-2-9B	14.97	12.87	32.50	24.51	14.55	11.51	39.56	29.92	25.40	19.70
Qwen2-7B Instruct	43.51	27.29	40.87	24.53	43.81	29.25	41.45	24.66	42.41	26.43
LLaMA-3-8B Instruct	39.48	32.00	39.65	25.10	40.23	32.48	42.64	28.03	40.50	29.40
Gemma-2-9B It	44.34	27.74	44.58	25.94	45.11	34.52	44.51	26.81	44.64	28.75
GLM-4-9B Chat	42.88	38.49	43.25	26.78	42.47	38.17	44.65	28.41	43.31	32.96
Mistral-7B Instruct	39.77	26.97	41.31	24.16	42.46	30.52	43.20	27.17	41.69	27.20
GPT-4o-Mini	46.96	44.92	46.46	42.86	47.08	42.90	46.89	43.71	46.85	43.60
GPT-4-Turbo	48.02	45.19	48.14	43.13	47.72	43.63	48.59	43.15	48.12	43.77

Table A.5: Balanced accuracy and Macro F1 of 10 LLMs on the benchmark using various prompts.

Methods	Zero-shot w/o CoT		Zero-shot w/ CoT		Few-shot w/o CoT		Few-shot w/ CoT		Overall	
	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
Qwen2-7B	58.64	31.82	62.33	49.40	59.71	45.39	64.49	54.80	61.29	45.35
LLaMA-3-8B	52.78	29.72	55.78	27.65	53.25	32.13	58.83	55.64	55.16	36.28
Gemma-2-9B	51.77	22.67	58.18	42.76	53.72	17.31	61.93	61.14	56.40	35.97
Qwen2-7B Instruct	65.11	47.85	66.63	41.27	66.79	44.88	66.67	41.01	66.30	43.75
LLaMA-3-8B Instruct	62.84	62.92	64.04	43.01	63.44	63.39	67.71	45.43	64.51	53.69
Gemma-2-9B It	70.37	43.53	69.03	41.08	71.10	44.81	70.67	43.03	70.29	43.11
GLM-4-9B Chat	64.82	52.56	68.24	42.58	65.38	52.97	68.44	47.10	66.72	48.80
Mistral-7B Instruct	62.68	50.90	65.58	37.33	63.29	60.13	66.40	43.80	64.49	48.04
GPT-4o-Mini	68.00	60.80	70.27	54.20	68.82	55.06	70.43	56.35	69.38	56.60
GPT-4-Turbo	68.76	60.67	71.27	53.31	69.80	56.01	71.35	53.18	70.29	55.79

Table A.6: Precision and recall of 10 LLMs on the benchmark using various prompts.

Task: Multi-hop Reasoning
Q: Did the 1952 Colgate Red Raiders football team win the game against the Buffalo Bulls football team, representing the University of Buffalo, on October 4?
A: No

```
| Date|Opponent|Site|Result |
|--------------|-----------------|--------------|---------|
| September 20 | at Cortland | Cortland, NY | L 13–33 |
| September 27 | Connecticut | | L 7–47 |
| October 4 | Colgate | | L 0–13 |
| October 11 | Lehigh | | L 7–26 |
| October 18 | Bucknell | | L 0–22 |
| October 25 | Western Reserve | | L 13–35 |
| November 1 | Alfred | | L 7–19 |
| November 8 | St. Lawrence | | W 12–6 |
| Date|Opponent|Site|Result |
|--------------|--------------------|-------------|---------|
| September 27 | Cornell | | W 14–7 |
| October 4 | Buffalo | Buffalo, NY | W 13–0 |
| October 11 | Rutgers | | W 13–7 |
| October 18 | Harvard | | L 20–21 |
| October 25 | Bucknell | | W 28–0 |
| November 1 | Mississippi College | | W 53–12 |
| November 8 | Holy Cross | | L 7–13 |
| November 15 | Syracuse | | L 14–20 |
| November 29 | Brown | | W 33–27 |
| 1952 Buffalo Bulls football |
|--------------|---------------|
| Conference | Independent |
| 1952 record | 1–7 |
| Head coach | |
| Home stadium | Civic Stadium |
| 1952 Colgate Red Raiders football |
|--------------|-----------------------|
| Conference | Independent |
| 1952 record | 6–3 |
| Head coach | |
| Captain | Donald Main |
| Home stadium | Colgate Athletic Field |
```

LLaMA-3-8B Instruct
To answer this question, we need to find the game between the 1952 Colgate Red Raiders football team and the 1952 Buffalo Bulls football team. We can do this by looking at the structured data provided.

The data shows that the Colgate Red Raiders football team played the following games:

```
| Date|Opponent|Site|Result |
|--------------|--------------------|-------------|---------|
| September 20 | at Cortland | Cortland, NY | L 13–33 |
| September 27 | Connecticut | | L 7–47 |
| October 4 | Colgate | | L 0–13 |
| October 11 | Lehigh | | L 7–26 |
| October 18 | Bucknell | | L 0–22 |
| October 25 | Western Reserve | | L 13–35 |
| November 1 | Alfred | | L 7–19 |
| November 8 | St. Lawrence | | W 12–6 |
```

The data also shows that the Buffalo Bulls football team played the following games:

```
| Date|Opponent|Site|Result |
|--------------|--------------------|-------------|---------|
| September 27 | Cornell | | W 14–7 |
| October 4 | Buffalo | Buffalo, NY | W 13–0 |
| October 11 | Rutgers | | W 13–7 |
| October 18 | Harvard | | L 20–21 |
| October 25 | Bucknell | | W 28–0 |
| November 1 | Mississippi College | | W 53–12 |
| November 8 | Holy Cross | | L 7–13 |
| November 15 | Syracuse | | L 14–20 |
| November 29 | Brown | | W 33–27 |
```

By comparing the dates and opponents, we can see that the game between the Colgate Red Raiders football team and the Buffalo Bulls football team was not played on October 4. The Colgate Red Raiders football team played Colgate on October 4, not the Buffalo Bulls.
A: Yes

Gemma-2-9B It
Let's look at the provided data:
* **Colgate's Schedule:** The data shows Colgate played Buffalo on October 4th, 1952.
* **Colgate's Result:** The result for that game is listed as "W 13–0".
Therefore, the 1952 Colgate Red Raiders football team won the game against the Buffalo Bulls.
A: Yes

GPT-4o-turbo
The data provided shows the results of the games played by both the Colgate Red Raiders and the Buffalo Bulls during the 1952 football season. On October 4, the game between Colgate and Buffalo is specifically mentioned. According to the data, the result of the game was a win for Buffalo with a score of 13-0 over Colgate.
Therefore, the 1952 Colgate Red Raiders football team did not win against the Buffalo Bulls football team on October 4, 1952.
A: No

Figure A.4: Case Study of Multi-hop Reasoning.

Task: Composition Understanding
Q: Was The Sounds of the Sounds of Science given an A rating by Stylus?
A: No

```
|   Review scores   |
|-----------------|-------------|
| Source          | Rating      |
| Allmusic        |             |
| Pitchfork Media | (8.0/10)    |
| Stylus          | (B)         |
| Dusted Magazine | (favorable) |
```

Gemma-2-9B
We don't have enough information about the review score for The Sounds of the Sounds of Science by Stylus. The given data only includes a rating of (B) for Stylus. Therefore, we cannot determine if the review score was A or not.
A: NEI

Gemma-2-9B It
Based on the provided data, Stylus gave The Sounds of the Sounds of Science a rating of (B). There is no information suggesting that it received an A rating from Stylus.
A: No

GPT-4o-mini
To determine if "The Sounds of the Sounds of Science" was given an A rating by Stylus, we need to look at the structured data provided. The relevant entry for Stylus shows a rating of (B). Since an A rating is higher than a B rating, it implies that the album did not receive an A rating from Stylus.
A: No

Figure A.5: Case Study of Composition Understanding.

Task: Combining Structured and Unstructured
Q: Are there two varieties of Sorbus hupehensis with narrow leaflets (Sorbus hupehensis var. hupehensis) and broad leaflets (Sorbus hupehensis var. paucijuga), even though its leaves are colored green on above and beneath are paler?
A: Yes

```
|   Sorbus hupehensis      |
|------------------------------|--------------|
| Scientific classification |              |
| Kingdom:                  | Plantae      |
| (unranked):               | Angiosperms  |
| (unranked):               | Eudicots     |
| (unranked):               | Rosids       |
| Order:                    | Rosales      |
| Family:                   | Rosaceae     |
| Genus:                    | Sorbus       |
| Subgenus:                 | Sorbus       |
| Species:                  | S. hupehensis |
| Binomial name             |              |
```

* Sorbus hupehensis var. hupehensis. Leaves with 9-17 narrow leaflets. Most of the species' range.
* Sorbus hupehensis var. paucijuga (D.K.Zang & P.C.Huang) L.T.Lu. Leaves with 7-9 broad leaflets. Shandong Province.

LLaMA-3-8B
There are two varieties of Sorbus hupehensis with narrow leaflets (Sorbus hupehensis var. hupehensis) and broad leaflets (Sorbus hupehensis var. paucijuga), even though its leaves are colored green on above and beneath are paler.
A: NEI

LLaMA-3-8B Instruct
The question asks if there are two varieties of Sorbus hupehensis with narrow leaflets and broad leaflets. According to the provided data, we see that Sorbus hupehensis var. hupehensis has leaves with 9-17 narrow leaflets, while Sorbus hupehensis var. paucijuga has leaves with 7-9 broad leaflets. This suggests that there are indeed two varieties of Sorbus hupehensis with distinct leaf characteristics.

However, the question also mentions that the leaves of Sorbus hupehensis are colored green on above and beneath, but are paler beneath. This information does not provide any contradictory evidence to the existence of the two varieties. Therefore, based on the provided data, I conclude that the answer is...
A: Yes

GPT-4o-turbo
The question asks whether there are two varieties of Sorbus hupehensis characterized by different leaf structures, specifically narrow and broad leaflets. The provided data distinctly mentions two varieties:
1. Sorbus hupehensis var. hupehensis, which has 9-17 narrow leaflets.
2. Sorbus hupehensis var. paucijuga, which has 7-9 broad leaflets.

The question also mentions the color of the leaves (green above and paler beneath), but this characteristic does not affect the classification of the varieties based on leaf structure. Therefore, the color of the leaves is irrelevant to determining the existence of the two varieties based on the leaflet width.

Based on the provided data, it is clear that there are indeed two varieties of Sorbus hupehensis, each defined by the width of the leaflets as described.
A: Yes

Figure A.6: Case Study of Combining Structured and Unstructured.

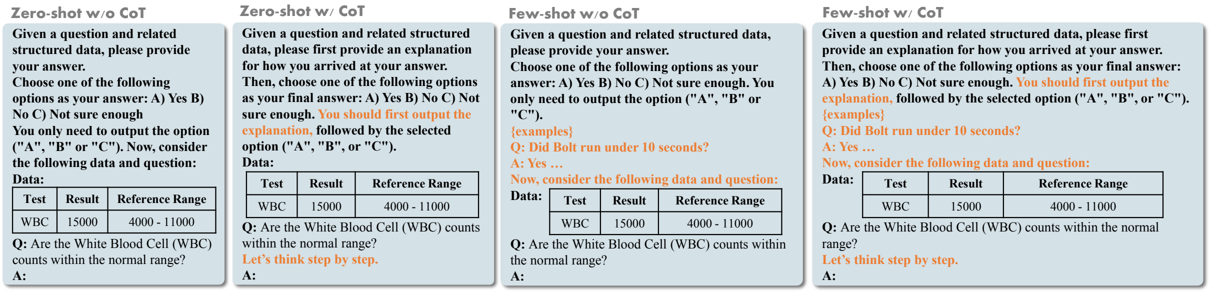


Figure A.7: Prompts used in different settings (main differences with zero-shot w/o CoT are marked in orange).

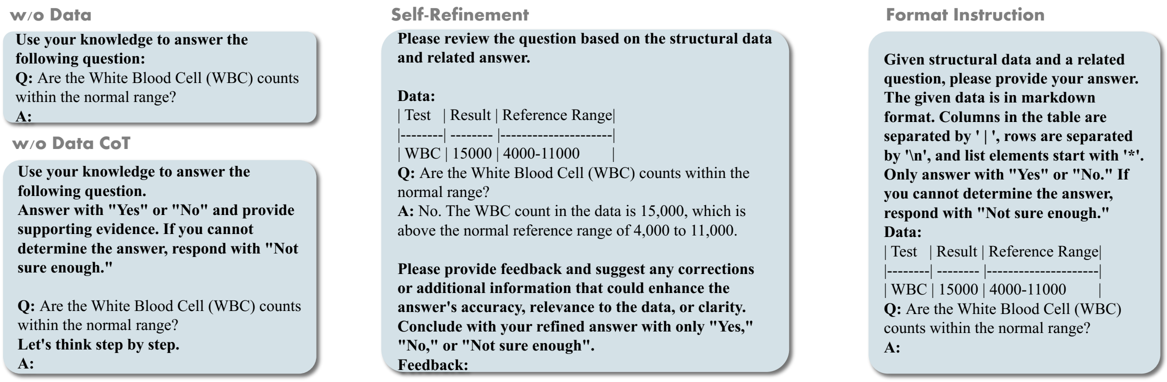


Figure A.8: Examples of different prompting strategies.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
Qwen2-7B	58.11	28.30	55.32	28.24	68.87	34.05	65.94	38.31	69.78	41.26
LLaMA-3-8B	54.34	28.48	49.83	28.36	63.72	29.61	53.61	34.08	63.42	34.00
Gemma-2-9B	51.77	15.98	48.47	21.45	67.26	30.55	60.43	25.87	61.55	25.92
Qwen2-7B Instruct	61.48	54.58	66.00	40.52	71.62	47.91	70.20	57.46	73.95	53.33
LLaMA-3-8B Instruct	57.41	62.28	62.58	54.78	69.43	70.61	61.54	60.94	72.15	73.28
Gemma-2-9B It	60.51	51.36	73.96	33.03	81.38	44.73	71.58	59.95	82.41	52.46
GLM-4-9B Chat	63.06	59.27	63.75	46.70	70.01	50.67	67.78	63.93	73.41	56.80
Mistral-7B Instruct	61.48	55.37	59.07	43.44	70.98	52.03	62.60	54.98	73.77	59.07
GPT-4o-Mini	64.51	62.52	67.79	60.13	75.06	58.04	68.80	67.42	75.06	62.10
GPT-4-Turbo	64.16	61.76	68.71	61.93	75.03	54.90	70.78	70.15	77.18	61.59
Overall	59.68	47.99	61.55	41.86	71.34	47.31	65.33	53.31	72.27	51.98

Table A.7: Precision and recall of 10 LLMs on the benchmark across five factual tasks under the zero-shot w/o CoT setting.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	BA	MF1	BA	MF1	BA	MF1	BA	MF1	BA	MF1
Qwen2-7B	19.42	18.07	23.62	19.35	22.30	19.48	27.85	23.99	27.13	21.99
LLaMA-3-8B	20.35	17.12	21.08	17.39	18.61	16.23	32.55	21.59	22.01	18.34
Gemma-2-9B	11.08	10.58	15.57	12.84	18.50	14.40	16.96	14.42	15.33	12.74
Qwen2-7B Instruct	40.04	30.66	44.59	29.03	43.23	25.79	40.16	34.04	44.43	29.72
LLaMA-3-8B Instruct	37.14	35.27	39.92	31.00	38.34	37.48	38.26	37.28	40.23	39.96
Gemma-2-9B It	38.61	32.89	44.50	29.55	45.61	33.09	55.90	37.33	48.18	35.80
GLM-4-9B Chat	38.22	36.99	44.85	37.61	40.91	34.71	41.31	42.14	44.53	38.48
Mistral-7B Instruct	38.64	26.56	38.69	25.59	39.45	25.58	34.91	30.13	41.78	28.08
GPT-4o-Mini	41.68	41.14	49.20	46.20	46.31	41.15	52.58	52.69	46.86	42.68
GPT-4-Turbo	41.05	40.17	50.43	47.09	47.20	39.42	58.31	60.05	49.42	43.67
Overall	32.62	28.95	37.25	29.56	36.05	28.73	39.88	35.37	37.99	31.15

Table A.8: Balanced accuracy and Macro F1 of 10 LLMs on the benchmark across five factual tasks under the zero-shot w/o CoT setting.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Qwen2-7B	57.65	57.56	39.63	42.68	52.11	59.05	59.21	59.50	55.22	61.00
LLaMA-3-8B	28.50	37.79	24.15	31.16	28.98	39.88	30.10	38.53	32.33	43.38
Gemma-2-9B	45.35	50.28	37.92	42.52	44.35	52.43	44.03	48.90	47.74	54.98
Qwen2-7B Instruct	53.92	56.87	31.66	39.43	39.74	49.90	51.24	55.54	45.18	55.02
LLaMA-3-8B Instruct	50.37	54.01	36.13	42.62	40.61	49.98	51.49	54.39	49.30	57.93
Gemma-2-9B It	48.75	53.66	33.91	43.30	40.61	51.25	57.46	60.31	44.87	55.46
GLM-4-9B Chat	53.57	58.14	35.52	45.05	39.00	49.17	56.22	59.62	45.09	55.43
Mistral-7B Instruct	43.93	50.87	32.00	40.89	34.94	44.83	50.00	55.88	40.95	51.28
GPT-4o-Mini	59.10	61.50	50.89	59.04	51.64	60.06	65.18	66.46	56.22	64.12
GPT-4-Turbo	58.44	61.04	51.71	60.26	49.48	57.78	64.93	65.93	52.64	61.56
Overall	49.96	54.17	37.35	44.70	42.15	51.43	52.99	56.51	46.95	56.02

Table A.9: Accuracy and F1 score of 10 LLMs on the benchmark across five factual tasks under the zero-shot w/ CoT setting.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
Qwen2-7B	60.38	57.65	61.86	39.63	70.62	52.11	66.14	59.21	71.26	55.22
LLaMA-3-8B	57.51	28.50	52.43	24.15	66.28	28.98	57.02	30.10	67.36	32.33
Gemma-2-9B	58.46	45.35	55.55	37.92	67.74	44.35	57.02	44.03	69.66	47.74
Qwen2-7B Instruct	61.87	53.92	69.06	31.66	73.41	39.74	67.39	51.24	75.92	45.18
LLaMA-3-8B Instruct	60.99	50.37	63.89	36.13	69.48	40.61	63.20	51.49	74.34	49.30
Gemma-2-9B It	61.11	48.75	71.34	33.91	77.50	40.61	66.14	57.46	78.25	44.87
GLM-4-9B Chat	64.25	53.57	69.49	35.52	74.36	39.00	65.64	56.22	76.37	45.09
Mistral-7B Instruct	62.24	43.93	64.62	32.00	72.72	34.94	65.04	50.00	74.37	40.95
GPT-4o-Mini	64.24	59.10	72.71	50.89	76.01	51.64	68.24	65.18	76.66	56.22
GPT-4-Turbo	64.22	58.44	74.82	51.71	76.52	49.48	67.30	64.93	77.91	52.64
Overall	61.53	49.96	65.58	37.35	72.46	42.15	64.31	52.99	74.21	46.95

Table A.10: Precision and recall of 10 LLMs on the benchmark across five factual tasks under the zero-shot w/ CoT setting.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	BA	MF1	BA	MF1	BA	MF1	BA	MF1	BA	MF1
Qwen2-7B	36.98	26.80	40.23	23.88	39.31	26.03	37.19	28.95	42.11	25.92
LLaMA-3-8B	19.80	18.25	18.81	17.00	17.94	16.90	18.92	18.46	20.44	19.13
Gemma-2-9B	31.21	24.49	33.22	23.59	31.07	23.50	28.77	24.23	34.96	25.27
Qwen2-7B Instruct	36.21	26.54	41.80	22.07	41.48	23.18	35.79	27.41	43.30	23.86
LLaMA-3-8B Instruct	34.25	25.21	41.18	23.77	37.35	22.84	32.38	26.36	45.42	26.09
Gemma-2-9B It	36.29	25.70	46.34	24.21	46.42	25.20	51.45	35.83	46.82	25.81
GLM-4-9B Chat	38.35	28.25	42.97	24.81	45.16	24.60	47.18	31.43	46.62	26.48
Mistral-7B Instruct	36.19	24.77	40.53	22.61	43.70	22.41	36.69	28.33	45.62	24.15
GPT-4o-Mini	40.26	40.38	48.84	42.82	46.16	40.34	46.80	46.44	47.69	41.38
GPT-4-Turbo	42.68	41.29	50.38	43.69	48.17	39.82	53.91	49.44	48.18	41.04
Overall	35.22	28.17	40.43	26.84	39.68	26.48	38.91	31.69	42.12	27.91

Table A.11: Balanced accuracy and Macro F1 of 10 LLMs on the benchmark across five factual tasks under the zero-shot w/ CoT setting.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Qwen2-7B	50.19	54.03	39.36	44.78	46.28	54.34	50.75	53.96	50.23	57.83
LLaMA-3-8B	30.30	37.95	29.45	34.63	36.58	46.16	30.35	37.67	35.85	45.21
Gemma-2-9B	17.43	23.45	15.98	20.19	18.65	25.42	18.41	25.58	18.46	24.95
Qwen2-7B Instruct	54.35	57.82	36.45	43.11	44.58	53.43	57.21	60.90	49.14	57.50
LLaMA-3-8B Instruct	63.37	58.22	55.42	53.08	70.17	68.47	63.68	60.77	73.30	72.09
Gemma-2-9B It	55.12	56.73	34.22	41.49	45.62	55.37	63.19	64.72	50.75	60.11
GLM-4-9B Chat	60.39	59.73	44.69	49.17	53.96	60.11	63.93	63.60	58.41	64.52
Mistral-7B Instruct	61.35	60.78	52.10	52.54	66.71	68.99	63.68	61.93	68.38	70.15
GPT-4o-Mini	60.38	62.24	52.50	58.28	52.55	59.80	66.42	67.15	54.84	62.19
GPT-4-Turbo	60.38	61.88	55.38	61.06	51.63	58.24	66.42	66.91	54.98	63.15
Overall	51.33	53.28	41.55	45.83	48.67	55.03	54.40	56.32	51.43	57.77

Table A.12: Accuracy and F1 score of 10 LLMs on the benchmark across five factual tasks under the few-shot w/o CoT setting.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
Qwen2-7B	59.91	50.19	57.40	39.36	68.94	46.28	61.30	50.75	69.95	50.23
LLaMA-3-8B	53.94	30.30	49.07	29.45	66.43	36.58	55.05	30.35	65.39	35.85
Gemma-2-9B	54.89	17.43	48.64	15.98	64.74	18.65	65.32	18.41	66.72	18.46
Qwen2-7B Instruct	64.18	54.35	67.43	36.45	73.35	44.58	71.91	57.21	74.64	49.14
LLaMA-3-8B Instruct	58.19	63.37	63.33	55.42	68.70	70.17	64.32	63.68	73.42	73.30
Gemma-2-9B It	62.17	55.12	73.25	34.22	80.08	45.62	71.86	63.19	81.23	50.75
GLM-4-9B Chat	62.74	60.39	65.13	44.69	70.68	53.96	70.20	63.93	74.36	58.41
Mistral-7B Instruct	61.39	61.35	61.66	52.10	71.83	66.71	65.70	63.68	73.17	68.38
GPT-4o-Mini	65.42	60.38	67.54	52.50	76.16	52.55	68.91	66.42	76.72	54.84
GPT-4-Turbo	64.26	60.38	70.50	55.38	76.40	51.63	67.73	66.42	78.20	54.98
Overall	60.71	51.33	62.40	41.55	71.73	48.67	66.23	54.40	73.38	51.43

Table A.13: Precision and recall of 10 LLMs on the benchmark across five factual tasks under the few-shot w/o CoT setting.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	BA	MF1	BA	MF1	BA	MF1	BA	MF1	BA	MF1
Qwen2-7B	36.48	26.20	34.27	24.48	35.19	24.32	57.21	30.75	36.01	25.16
LLaMA-3-8B	23.39	18.58	26.25	19.28	26.55	20.46	22.82	18.56	25.00	19.54
Gemma-2-9B	14.01	11.48	14.54	10.91	14.74	11.74	12.36	12.65	15.24	11.31
Qwen2-7B Instruct	42.22	31.10	43.03	32.05	43.52	32.24	54.20	42.54	45.12	32.96
LLaMA-3-8B Instruct	36.66	35.01	41.74	31.90	37.46	36.98	40.16	39.54	41.87	42.13
Gemma-2-9B It	40.02	35.09	45.79	31.18	46.99	34.38	61.71	47.96	44.61	35.25
GLM-4-9B Chat	38.28	37.16	44.50	36.40	39.18	35.47	44.21	43.11	45.03	39.22
Mistral-7B Instruct	39.92	35.92	43.07	29.12	42.08	34.00	43.79	43.40	43.94	34.62
GPT-4o-Mini	42.76	41.79	47.27	42.27	46.73	39.86	55.97	53.10	49.30	40.46
GPT-4-Turbo	41.90	41.22	49.54	44.33	48.96	39.64	45.09	44.99	47.29	41.07
Overall	35.56	31.36	39.00	30.19	38.14	30.91	43.75	37.66	39.34	32.17

Table A.14: Balanced accuracy and Macro F1 of 10 LLMs on the benchmark across five factual tasks under the few-shot w/o CoT setting.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Qwen2-7B	57.98	58.50	47.25	49.93	58.91	64.52	61.19	62.20	61.66	66.44
LLaMA-3-8B	56.03	55.56	48.52	46.85	60.50	63.71	54.97	54.24	65.22	66.91
Gemma-2-9B	60.02	58.96	53.72	52.45	68.17	69.11	62.94	62.50	70.94	71.31
Qwen2-7B Instruct	51.68	55.78	31.88	39.99	41.24	51.29	52.74	58.31	44.62	54.55
LLaMA-3-8B Instruct	52.21	57.44	40.64	49.51	42.44	52.08	53.23	57.34	48.74	58.37
Gemma-2-9B It	53.86	57.52	34.86	44.45	42.03	52.95	61.44	63.33	45.07	55.99
GLM-4-9B Chat	56.36	59.70	39.08	47.86	46.19	55.65	60.45	63.31	51.28	60.48
Mistral-7B Instruct	48.49	54.83	38.30	46.84	43.53	52.71	58.21	61.53	48.05	57.49
GPT-4o-Mini	62.36	63.52	52.88	60.49	53.66	61.60	70.15	70.40	57.11	64.72
GPT-4-Turbo	60.03	62.38	50.98	59.20	48.95	57.88	66.42	67.97	51.70	61.01
Overall	55.90	58.42	43.81	49.76	50.56	58.15	60.17	62.11	54.44	61.73

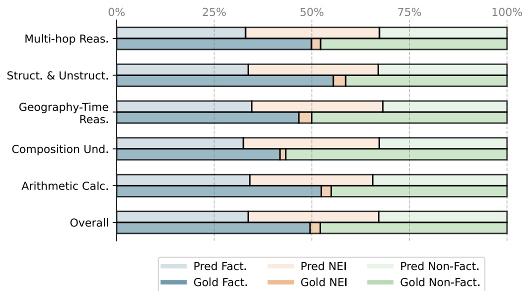
Table A.15: Accuracy and F1 score of 10 LLMs on the benchmark across five factual tasks under the few-shot w/ CoT setting.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
Qwen2-7B	61.01	57.98	63.75	47.25	72.32	58.91	69.20	61.19	73.67	61.66
LLaMA-3-8B	56.82	56.03	59.04	48.52	67.47	60.50	60.35	54.97	69.66	65.22
Gemma-2-9B	58.77	60.02	61.26	53.72	70.30	68.17	65.10	62.94	72.25	70.94
Qwen2-7B Instruct	62.31	51.68	68.47	31.88	73.24	41.24	67.83	52.74	75.82	44.62
LLaMA-3-8B Instruct	64.28	52.21	68.66	40.64	73.38	42.44	63.49	53.23	75.91	48.74
Gemma-2-9B It	63.13	53.86	72.82	34.86	78.73	42.03	67.84	61.44	79.02	45.07
GLM-4-9B Chat	63.87	56.36	68.98	39.08	75.07	46.19	69.23	60.45	76.90	51.28
Mistral-7B Instruct	64.16	48.49	65.57	38.30	72.79	43.53	65.79	58.21	74.63	48.05
GPT-4o-Mini	64.77	62.36	72.79	52.88	76.08	53.66	71.00	70.15	76.38	57.11
GPT-4-Turbo	65.18	60.03	72.94	50.98	78.59	48.95	70.02	66.42	78.80	51.70
Overall	62.43	55.90	67.43	43.81	73.80	50.56	66.99	60.17	75.30	54.44

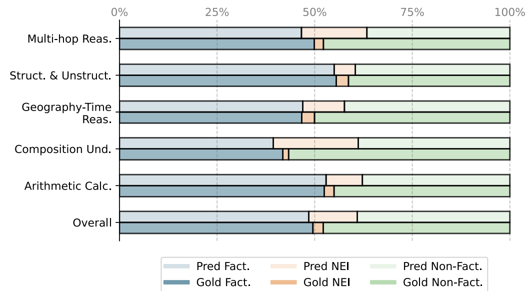
Table A.16: Precision and recall of 10 LLMs on the benchmark across five factual tasks under the few-shot w/ CoT setting.

Methods	Arithmetic Calc.		Spatiotemporal Cogn.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
	BA	MF1	BA	MF1	BA	MF1	BA	MF1	BA	MF1
Qwen2-7B	36.38	27.26	39.40	27.52	38.25	28.76	38.74	30.34	41.33	29.97
LLaMA-3-8B	33.82	25.66	36.34	25.64	33.19	26.29	44.65	29.37	36.41	28.65
Gemma-2-9B	36.45	27.61	40.59	28.89	38.58	29.72	47.59	33.37	41.84	32.09
Qwen2-7B Instruct	37.95	26.50	40.97	22.27	41.58	23.80	37.52	29.18	46.30	24.26
LLaMA-3-8B Instruct	36.26	27.89	45.20	27.38	42.27	25.74	34.38	28.20	45.90	27.54
Gemma-2-9B It	38.08	27.32	45.93	24.73	46.63	26.56	50.37	37.23	45.33	26.06
GLM-4-9B Chat	37.39	28.48	44.61	26.39	48.13	27.40	46.04	36.53	48.93	31.90
Mistral-7B Instruct	38.88	27.36	43.15	25.80	45.75	25.82	38.37	34.12	44.68	26.94
GPT-4o-Mini	41.51	41.51	49.39	43.89	47.84	41.21	46.62	47.22	45.37	41.44
GPT-4-Turbo	43.31	42.05	49.59	42.99	49.79	40.50	55.17	50.34	50.68	41.24
Overall	38.00	30.16	43.52	29.55	43.20	29.58	43.95	35.59	44.68	31.01

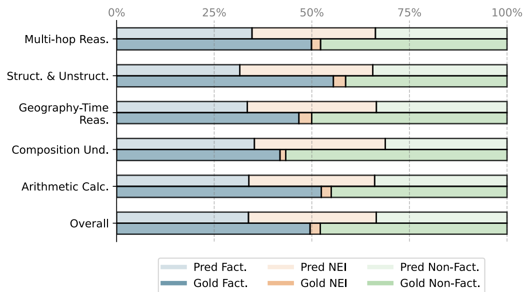
Table A.17: Balanced accuracy and Macro F1 of 10 LLMs on the benchmark across five factual tasks under the few-shot w/ CoT setting.



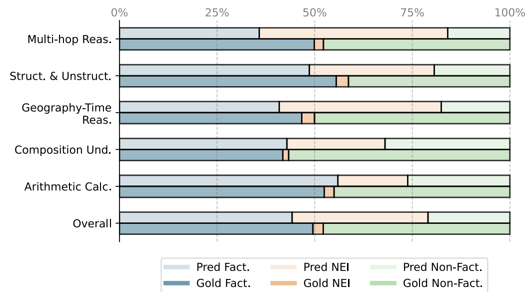
(a) GPT-4O Mini



(b) Llama3-8B Instruct



(c) Gemma2-9B It



(d) Qwen2-7B-Instruct

Figure A.9: Responses Distributions of Different Models.

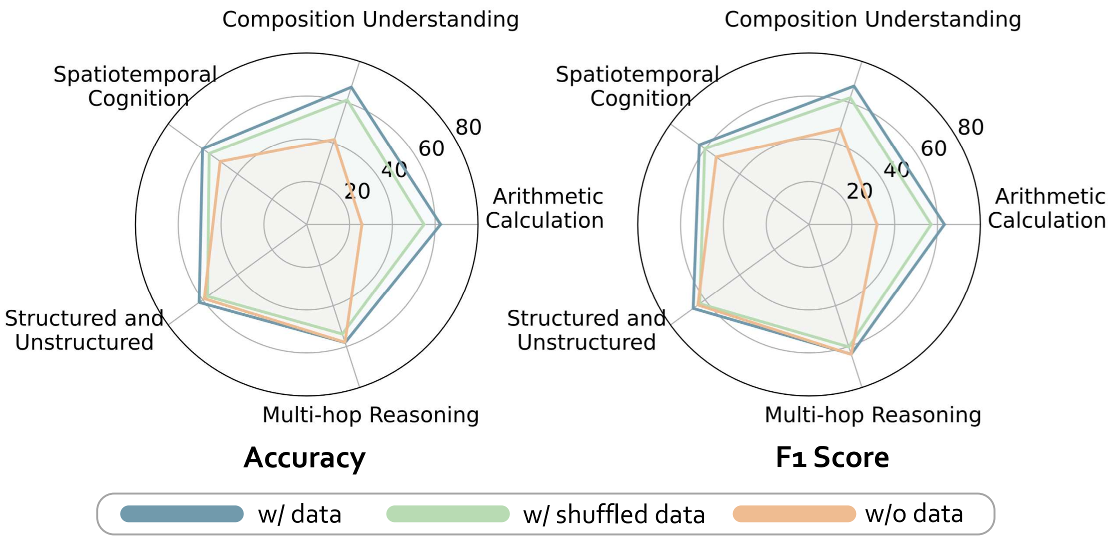
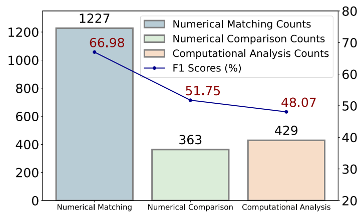
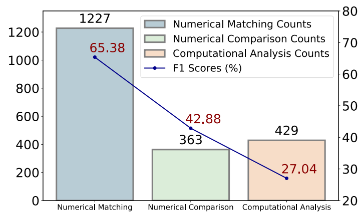


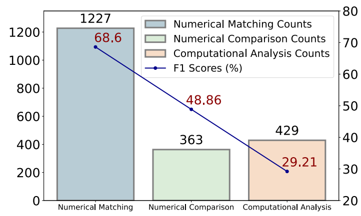
Figure A.10: Accuracy and F1 score of GPT-4o-mini under different settings of structured evidence.



(a) Llama3-8B Instruct

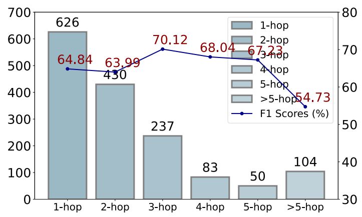


(b) Gemma2-9B It

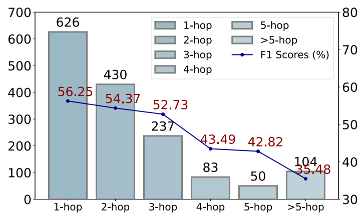


(c) Qwen2-7B-Instruct

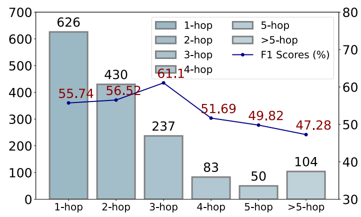
Figure A.11: Performance of Different Models on Arithmetic Calculation.



(a) Llama3-8B Instruct



(b) Gemma2-9B It

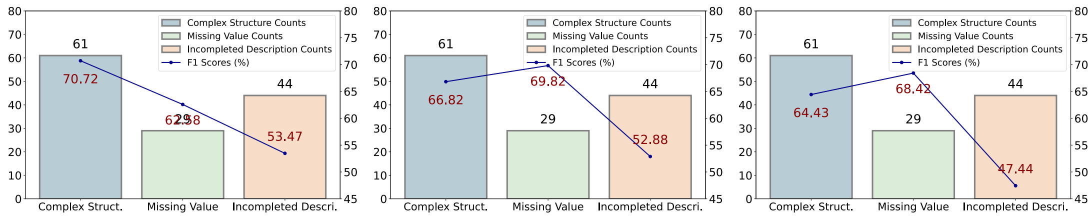


(c) Qwen2-7B-Instruct

Figure A.12: Performance of Different Models on Multi-hop Reasoning.

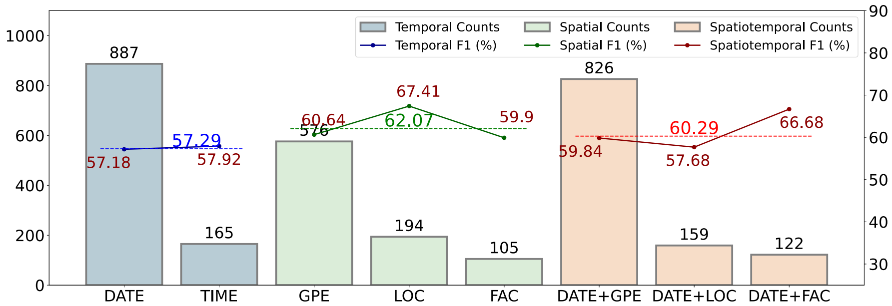
Table A.18: GPT-4o-mini's evidence resilience across different factual tasks under zero-shot settings without CoT prompts. The percentage of decrease with respect to the setting with original structured data (w/ data) is shown in brackets.

Settings	Metrics	Arithmetic Calc.		Geography-Time Reas.		Multi-hop Reas.		Composition Und.		Struct. & Unstruct.	
		Score (%)	Decr.(%)	Score	Decr.(%)	Score	Decr.(%)	Score	Decr.(%)	Score	Decr.(%)
w/ data	Acc	63.93	-	59.92	-	67.56	-	71.88	-	65.94	-
	F1	64.15	-	62.96	-	69.66	-	72.44	-	68.50	-
	Prec.	64.39	-	67.16	-	76.29	-	73.10	-	72.89	-
	Recall	63.93	-	59.92	-	67.56	-	71.88	-	65.94	-
w/ shuffled data (rows)	Acc	62.68	(-1.99)	58.39	(-2.57)	64.81	(-4.12)	70.62	(-1.75)	63.18	(-4.25)
	F1	63.43	(-1.17)	61.73	(-1.96)	67.12	(-3.66)	71.16	(-1.77)	66.38	(-3.04)
	Prec.	64.77	(+0.58)	66.44	(-1.09)	75.76	(-0.70)	71.87	(-1.66)	73.31	(+0.57)
	Recall	62.68	(-1.99)	58.39	(-2.57)	64.81	(-4.12)	70.62	(-1.75)	63.18	(-4.25)
w/ shuffled data (columns)	Acc	61.19	(-4.22)	58.35	(-2.63)	65.04	(-3.73)	68.50	(-4.73)	63.87	(-3.12)
	F1	61.94	(-3.41)	61.66	(-2.08)	67.52	(-3.03)	69.42	(-4.19)	66.82	(-2.46)
	Prec.	63.79	(-0.93)	66.31	(-1.28)	76.11	(-0.23)	70.62	(-3.39)	72.63	(-0.35)
	Recall	61.19	(-4.22)	58.35	(-2.63)	65.04	(-3.73)	68.50	(-4.73)	63.87	(-3.12)
w/ shuffled data (rows and columns)	Acc	54.78	(-14.31)	56.30	(-6.04)	58.86	(-12.88)	67.19	(-6.53)	59.25	(-10.15)
	F1	59.01	(-8.01)	57.21	(-9.13)	62.18	(-10.73)	67.26	(-7.15)	63.14	(-7.83)
	Prec.	60.98	(-5.30)	64.07	(-4.60)	73.09	(-4.20)	70.33	(-3.79)	70.64	(-3.09)
	Recall	56.78	(-11.18)	56.30	(-6.04)	59.86	(-11.40)	67.19	(-6.53)	58.25	(-11.66)
w/o data	Acc	25.89	(-38.04)	49.11	(-10.81)	53.33	(-14.23)	51.65	(-20.23)	55.43	(-10.51)
	F1	32.59	(-31.56)	52.85	(-10.11)	60.23	(-9.43)	57.79	(-14.65)	59.67	(-8.83)
	Prec.	51.25	(-13.14)	59.30	(-7.86)	70.38	(-5.91)	67.96	(-5.14)	66.36	(-6.53)
	Recall	25.89	(-38.04)	49.11	(-10.81)	53.33	(-14.23)	51.65	(-20.23)	55.43	(-10.51)

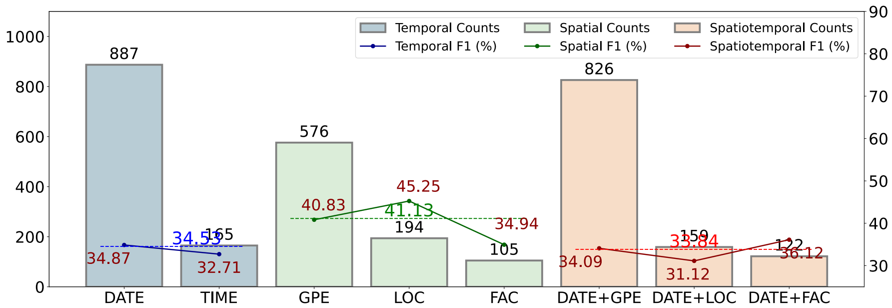


(a) Llama3-8B Instruct (b) Gemma2-9B It (c) Qwen2-7B-Instruct

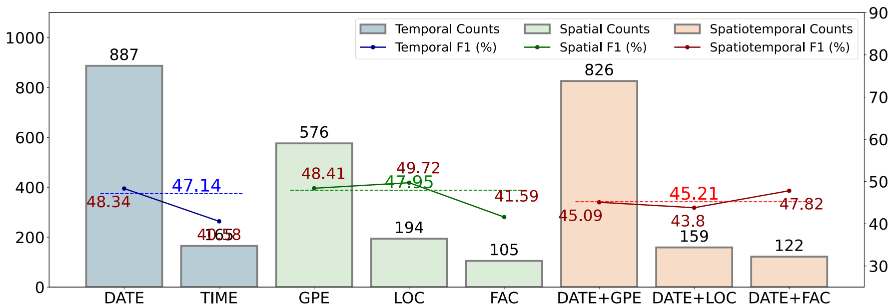
Figure A.13: Performance of Different Models on Composition Understanding.



(a) Llama3-8B Instruct

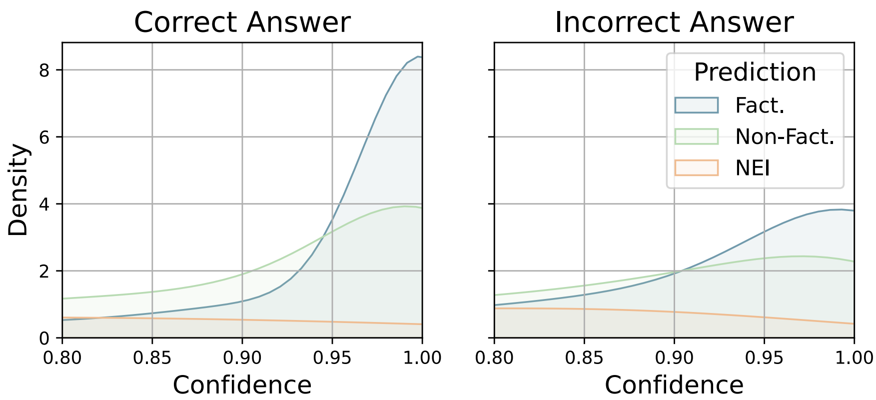


(b) Gemma2-9B It

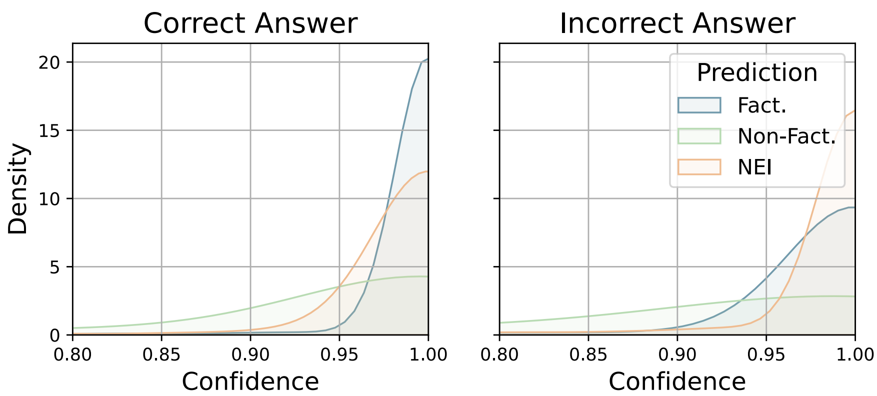


(c) Qwen2-7B-Instruct

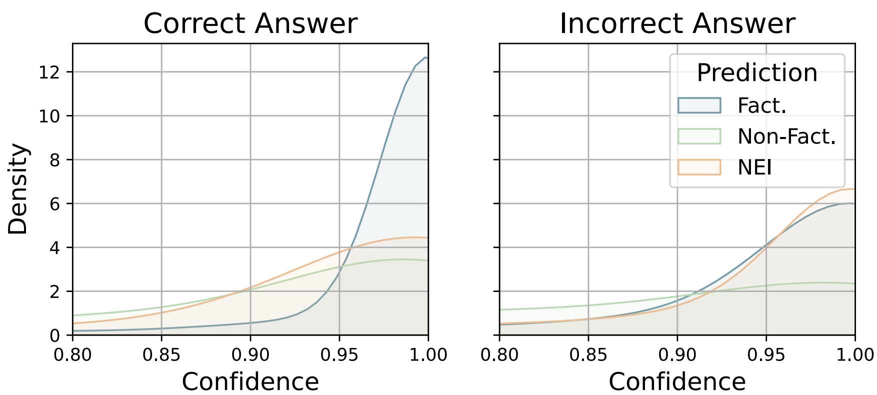
Figure A.14: Performance of Different Models on geography-time Reasoning.



(a) Llama3-8B Instruct



(b) Gemma2-9B It



(c) Qwen2-7B-Instruct

Figure A.15: Confidence of Different Models.

Appendix B

Supplementary materials for HyperG

B.1 Prompt Templates

This section provides supplementary examples of the templates used to prompt the LLMs in our experiments presented in Chapter 5.

TFV Prompt Template

A markdown table is provided below:

```
usa today all - usa high school basketball team
| player            | height | school                     | hometown       |
|:------------------|:-------|:---------------------------|:---------------|
| anthony randolph  | 6-10   | woodrow wilson high school | dallas, tx     |
| nolan smith       | 6-3    | oak hill academy           | washington, dc |
| corey fisher.     | 6-0    | st patrick high school     | elizabeth, nj  |
| nick calathes     | 6-4    | lake howell high school    | winter park, fl |
| austin freeman.   | 6-4    | dematha catholic high school | Hyattsville, md |
```

Here is a claim: anthony randolph be the tallest player

Based on all the available information, predict whether this claim is correct. Respond with 'Yes' or 'No' only.

Answer: Yes.

(a) An example prompt for the TFV task.

TQA Prompt Template

A markdown table is provided below:

```
2005 East Asian Games
| Rank | Nation                 | Gold | Silver | Bronze| Total |
|:-----|:-----------------------|-----:|-------:|------:|------:|
| 1    | China (CHN)            | 127  | 63.    | 33    | 223   |
| 2    | Japan (JPN)            | 46   | 56     | 77    | 179   |
| 3    | South Korea (KOR)      | 32   | 48.    | 65.   | 145   |
| 4    | Chinese Taipei (TPE)   | 12   | 34     | 26    | 72    |
| 5    | Macau (MAC)            | 11   | 16     | 17    | 44    |
| 6    | North Korea (PRK)      | 6    | 10.    | 20    | 36    |
| 7    | Hong Kong (HKG)        | 2    | 2      | 9     | 13    |
| 8    | Mongolia (MGL)         | 1    | 1.     | 6.    | 8.    |
| 9    | Guam (GUM)             | 0    | 0.     | 1.    | 1     |
| Total | Total                 | 237  | 230.   | 254   | 721.  |.
```

Here is a question: what was the only nation not to win a gold medal?.

Using all available information, provide a concise answer only. Do not include any reasoning or additional details.

Answer: Guam (GUM)

(b) An example prompt for the TQA task.

Figure B.1: Examples of the prompts used in the TFV and TQA experiments for HyperG proposed in this thesis.

Bibliography

[1] *Gemma: Open models based on gemini research and technology*, 2024.

[2] M. Akhtar, O. Cocarascu, and E. Simperl, *PubHealthTab: A public health table-based dataset for evidence-based fact checking*, in Findings of the Association for Computational Linguistics: NAACL 2022, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, eds., Seattle, United States, July 2022, Association for Computational Linguistics, pp. 1–16.

[3] R. Aly, Z. Guo, M. S. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, and A. Mittal, *Feverous: Fact extraction and verification over unstructured and structured information*, in Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1), 2021.

[4] I. Amazon Web Services, *What is structured data?*, 2024.

[5] M. An, F. Wu, C. Wu, K. Zhang, Z. Liu, and X. Xie, *Neural news recommendation with long-and short-term user representations*, in Proceedings of the 57th annual meeting of the association for computational linguistics, 2019, pp. 336–345.

[6] ——, *Neural news recommendation with long-and short-term user representations*, in Proceedings of the 57th annual meeting of the association for computational linguistics, 2019, pp. 336–345.

[7] S. Bai, F. Zhang, and P. H. Torr, *Hypergraph convolution and hypergraph attention*, Pattern Recognition, 110 (2021), p. 107637.

[8] A. Bellogín, P. Castells, and I. Cantador, *Statistical biases in information retrieval metrics for recommender systems*, Information Retrieval Journal, 20 (2017), pp. 606–634.

[9] K. R. Bhandari, S. Xing, S. Dan, and J. Gao, *On the robustness of language models for tabular question answering*, arXiv preprint arXiv:2406.12719, (2024).

[10] S. Bian, W. X. Zhao, K. Zhou, J. Cai, Y. He, C. Yin, and J.-R. Wen, *Contrastive curriculum learning for sequential user behavior modeling via data augmentation*, in Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21, New York, NY, USA, 2021, Association for Computing Machinery, p. 3737–3746.

[11] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, *Protein function prediction via graph kernels*, Bioinformatics, 21 (2005), pp. i47–i56.

[12] V. Borisov, T. Leemann, K. Sessler, J. Haug, M. Pawelczyk, and G. Kasneci, *Deep neural networks and tabular data: A survey*, IEEE transactions on neural networks and learning systems, (2022).

[13] V. Borisov, T. Leemann, K. Sessler, J. Haug, M. Pawelczyk, and G. Kasneci, *Deep neural networks and tabular data: A survey*, IEEE Transactions on Neural Networks and Learning Systems, 35 (2024), pp. 7499–7519.

[14] W. Cai, F. Feng, Q. Wang, T. Yang, Z. Liu, and C. Xu, *A causal view for item-level effect of recommendation on user preference*, in Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM '23, New York, NY, USA, 2023, Association for Computing Machinery, p. 240–248.

[15] Z. Chai, T. Zhang, L. Wu, K. Han, X. Hu, X. Huang, and Y. Yang, *Graphllm: Boosting graph reasoning ability of large language model*, arXiv preprint arXiv:2310.05845, (2023).

[16] H. Chen, Y. Li, X. Sun, G. Xu, and H. Yin, *Temporal meta-path guided explainable recommendation*, in Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM '21, New York, NY, USA, 2021, Association for Computing Machinery, p. 1056–1064.

[17] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al., *Evaluating large language models trained on code*, arXiv preprint arXiv:2107.03374, (2021).

[18] N. CHEN, Y. LI, J. TANG, AND J. LI, *Graphwiz: An instruction-following language model for graph computational problems*, in Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24, New York, NY, USA, 2024, Association for Computing Machinery, p. 353–364.

[19] P. CHEN, S. SARKAR, L. LAUSEN, B. SRINIVASAN, S. ZHA, R. HUANG, AND G. KARYPIS, *Hytrel: Hypergraph-enhanced tabular data representation learning*, Advances in Neural Information Processing Systems, 36 (2024).

[20] S. CHEN, J. WU, Y. WANG, C. JI, T. XIE, D. COCIORVA, M. SHARPS, C. LEVASSEUR, AND H. BRUNZELL, *Tabdeco: A comprehensive contrastive framework for decoupled representations in tabular data*, arXiv preprint arXiv:2411.11148, (2024).

[21] W. CHEN, M.-W. CHANG, E. SCHLINGER, W. Y. WANG, AND W. W. COHEN, *Open question answering over tables and text*, in International Conference on Learning Representations, 2020.

[22] W. CHEN, H. WANG, J. CHEN, Y. ZHANG, H. WANG, S. LI, X. ZHOU, AND W. Y. WANG, *Tabfact : A large-scale dataset for table-based fact verification*, in International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, April 2020.

[23] W. CHEN, H. ZHA, Z. CHEN, W. XIONG, H. WANG, AND W. Y. WANG, *HybridQA: A dataset of multi-hop question answering over tabular and textual data*, in Findings of the Association for Computational Linguistics: EMNLP 2020, T. Cohn, Y. He, and Y. Liu, eds., Online, Nov. 2020, Association for Computational Linguistics, pp. 1026–1036.

[24] X. CHEN, Z. ZHAO, L. CHEN, D. ZHANG, J. JI, A. LUO, Y. XIONG, AND K. YU, *Websrc: A dataset for web-based structural reading comprehension*, arXiv preprint arXiv:2101.09465, (2021).

[25] Y. CHEN, W. YE, G. XV, C. LIN, AND X. ZHU, *Tccm: Time and content-aware causal model for unbiased news recommendation*, in Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23, New York, NY, USA, 2023, Association for Computing Machinery, p. 3778–3782.

[26] E. Chien, C. Pan, J. Peng, and O. Milenkovic, *You are allset: A multiset function framework for hypergraph neural networks*, in International Conference on Learning Representations.

[27] P. Colon-Hernandez, C. Havasi, J. Alonso, M. Huggins, and C. Breazeal, *Combining pre-trained language models and structured knowledge*, arXiv preprint arXiv:2101.12294, (2021).

[28] L. Cui, H. Li, K. Chen, L. Shou, and G. Chen, *Tabular data augmentation for machine learning: Progress and prospects of embracing generative ai*, arXiv preprint arXiv:2407.21523, (2024).

[29] G. De Francisci Morales, A. Gionis, and C. Lucchese, *From chatter to headlines: harnessing the real-time web for personalized news recommendation*, in Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12, New York, NY, USA, 2012, Association for Computing Machinery, p. 153–162.

[30] A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch, *Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity*, Journal of medicinal chemistry, 34 (1991), pp. 786–797.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 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), 2019, pp. 4171–4186.

[32] Y. Dong, W. Sawin, and Y. Bengio, *Hnhn: Hypergraph networks with hyperedge neurons*, arXiv preprint arXiv:2006.12278, (2020).

[33] X. Fang, W. Xu, F. A. Tan, J. Zhang, Z. Hu, Y. J. Qi, S. Nickleach, D. Socolinsky, S. Sengamedu, C. Faloutsos, et al., *Large language models (llms) on tabular data: Prediction, generation, and understanding-a survey*, (2024).

[34] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, *Hypergraph neural networks*, in Proceedings of the AAAI conference on artificial intelligence, vol. 33, 2019, pp. 3558–3565.

[35] F. GARCIN, C. DIMITRAKAKIS, AND B. FALTINGS, *Personalized news recommendation with context trees*, in Proceedings of the 7th ACM Conference on Recommender Systems, 2013, pp. 105–112.

[36] C. GARDENT, A. SHIMORINA, S. NARAYAN, AND L. PEREZ-BELTRACHINI, *The WebNLG challenge: Generating text from RDF data*, in Proceedings of the 10th International Conference on Natural Language Generation, J. M. Alonso, A. Bugarín, and E. Reiter, eds., Santiago de Compostela, Spain, Sept. 2017, Association for Computational Linguistics, pp. 124–133.

[37] A. GHOSH, B. V. SAHITH, N. GANGULY, P. GOYAL, AND M. SINGH, *How robust are the tabular qa models for scientific tables? a study using customized dataset*, arXiv preprint arXiv:2404.00401, (2024).

[38] T. GLM, A. ZENG, B. XU, B. WANG, C. ZHANG, D. YIN, D. ROJAS, G. FENG, H. ZHAO, H. LAI, H. YU, H. WANG, J. SUN, J. ZHANG, J. CHENG, J. GUI, J. TANG, J. ZHANG, J. LI, L. ZHAO, L. WU, L. ZHONG, M. LIU, M. HUANG, P. ZHANG, Q. ZHENG, R. LU, S. DUAN, S. ZHANG, S. CAO, S. YANG, W. L. TAM, W. ZHAO, X. LIU, X. XIA, X. ZHANG, X. GU, X. LV, X. LIU, X. LIU, X. YANG, X. SONG, X. ZHANG, Y. AN, Y. XU, Y. NIU, Y. YANG, Y. LI, Y. BAI, Y. DONG, Z. QI, Z. WANG, Z. YANG, Z. DU, Z. HOU, AND Z. WANG, *Chatglm: A family of large language models from glm-130b to glm-4 all tools*, 2024.

[39] J. GUO, L. DU, AND H. LIU, *Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking*, arXiv preprint arXiv:2305.15066, (2023).

[40] V. GUPTA, M. MEHTA, P. NOKHIZ, AND V. SRIKUMAR, *INFOTABS: Inference on tables as semi-structured data*, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds., Online, July 2020, Association for Computational Linguistics, pp. 2309–2324.

[41] S. HAN, J. YOON, S. O. ARIK, AND T. PFISTER, *Large language models can automatically engineer features for few-shot tabular learning*, in Forty-first International Conference on Machine Learning.

[42] X. HE, K. DENG, X. WANG, Y. LI, Y. ZHANG, AND M. WANG, *Lightgcn: Simplifying and powering graph convolution network for recommendation*, in Pro-

ceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 639–648.

[43] S. HEGSELMANN, A. BUENDIA, H. LANG, M. AGRAWAL, X. JIANG, AND D. SONTAG, *Tabllm: Few-shot classification of tabular data with large language models*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2023, pp. 5549–5581.

[44] D. HENDRYCKS, C. BURNS, S. KADAVATH, A. ARORA, S. BASART, E. TANG, D. SONG, AND J. STEINHARDT, *Measuring mathematical problem solving with the math dataset*, in Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021.

[45] ——, *Measuring mathematical problem solving with the math dataset*, in Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021.

[46] J. HERZIG, T. MÜLLER, S. KRICHENE, AND J. M. EISENSCHLOS, *Open domain question answering over tables via dense retrieval*, arXiv preprint arXiv:2103.12011, (2021).

[47] X. HO, A.-K. DUONG NGUYEN, S. SUGAWARA, AND A. AIZAWA, *Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps*, in Proceedings of the 28th International Conference on Computational Linguistics, D. Scott, N. Bel, and C. Zong, eds., Barcelona, Spain (Online), Dec. 2020, International Committee on Computational Linguistics, pp. 6609–6625.

[48] N. HOLLMANN, S. MÜLLER, L. PURUCKER, ET AL., *Accurate predictions on small data with a tabular foundation model*, Nature, 637 (2025), pp. 319–326.

[49] E. J. HU, Y. SHEN, P. WALLIS, Z. ALLEN-ZHU, Y. LI, S. WANG, L. WANG, AND W. CHEN, *LoRA: Low-rank adaptation of large language models*, in International Conference on Learning Representations, 2022.

[50] L. HU, S. XU, C. LI, C. YANG, C. SHI, N. DUAN, X. XIE, AND M. ZHOU, *Graph neural news recommendation with unsupervised preference disentanglement*, in Proceedings of the 58th annual meeting of the association for computational linguistics, 2020, pp. 4255–4264.

[51] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, *Open graph benchmark: datasets for machine learning on graphs*, in Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Red Hook, NY, USA, 2020, Curran Associates Inc.

[52] X. Hu, J. Chen, X. Li, Y. Guo, L. Wen, S. Y. Philip, and Z. Guo, *Towards understanding factual knowledge of large language models*, in The Twelfth International Conference on Learning Representations, 2023.

[53] X. Huang, K. Han, Y. Yang, D. Bao, Q. Tao, Z. Chai, and Q. Zhu, *Can gnn be good adapter for llms?*, in Proceedings of the ACM Web Conference 2024, WWW '24, New York, NY, USA, 2024, Association for Computing Machinery, p. 893–904.

[54] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, *Tabtransformer: Tabular data modeling using contextual embeddings*, arXiv preprint arXiv:2012.06678, (2020).

[55] IBM, *Structured versus unstructured data: What's the difference?*, 2024.

[56] M. Iyyer, W.-t. Yih, and M.-W. Chang, *Search-based neural structured learning for sequential question answering*, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1821–1831.

[57] S. Jaitly, T. Shah, A. Shugani, and R. S. Grewal, *Towards better serialization of tabular data for few-shot classification*, arXiv preprint arXiv:2312.12464, (2023).

[58] J. Jiang, K. Zhou, Z. Dong, K. Ye, X. Zhao, and J.-R. Wen, *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, H. Bouamor, J. Pino, and K. Bali, eds., Singapore, Dec. 2023, Association for Computational Linguistics, pp. 9237–9251.

[59] B. Jin, C. Gao, X. He, D. Jin, and Y. Li, *Multi-behavior recommendation with graph convolutional networks*, in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 659–668.

[60] R. JIN, Y. LI, G. QI, N. HU, Y.-F. LI, J. CHEN, J. WANG, Y. CHEN, AND D. MIN, *Hgt: Leveraging heterogeneous graph-enhanced large language models for few-shot complex table understanding*, arXiv preprint arXiv:2403.19723, (2024).

[61] J. KIM, S. PARK, Y. KWON, Y. JO, J. THORNE, AND E. CHOI, *Factkg: Fact verification via reasoning on knowledge graphs*, in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 16190–16206.

[62] T. KOJIMA, S. S. GU, M. REID, Y. MATSUO, AND Y. IWASAWA, *Large language models are zero-shot reasoners*, Advances in neural information processing systems, 35 (2022), pp. 22199–22213.

[63] K. KONG, J. ZHANG, Z. SHEN, B. SRINIVASAN, C. LEI, C. FALOUTSOS, H. RANGWALA, AND G. KARYPIS, *Opentab: Advancing large language models as open-domain table reasoners*, in The Twelfth International Conference on Learning Representations.

[64] N. KOTONYA AND F. TONI, *Explainable automated fact-checking for public health claims*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), B. Webber, T. Cohn, Y. He, and Y. Liu, eds., Online, Nov. 2020, Association for Computational Linguistics, pp. 7740–7754.

[65] S. KUMAR, X. ZHANG, AND J. LESKOVEC, *Predicting dynamic embedding trajectory in temporal interaction networks*, in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 1269–1278.

[66] S. KWEON, Y. KWON, S. CHO, Y. JO, AND E. CHOI, *Open-WikiTable : Dataset for open domain question answering with complex reasoning over table*, in Findings of the Association for Computational Linguistics: ACL 2023, A. Rogers, J. Boyd-Graber, and N. Okazaki, eds., Toronto, Canada, July 2023, Association for Computational Linguistics, pp. 8285–8297.

[67] W. KWON, Z. LI, S. ZHUANG, Y. SHENG, L. ZHENG, C. H. YU, J. E. GONZALEZ, H. ZHANG, AND I. STOICA, *Efficient memory management for large language model serving with pagedattention*, 2023.

[68] H. LI, J. SU, Y. CHEN, Q. LI, AND Z. ZHANG, *Sheetcopilot: bringing software productivity to the next level through large language models*, in Proceedings of

the 37th International Conference on Neural Information Processing Systems, 2023, pp. 4952–4984.

[69] H. LI, J. SU, Y. CHEN, Q. LI, AND Z.-X. ZHANG, *Sheetcopilot: Bringing software productivity to the next level through large language models*, Advances in Neural Information Processing Systems, 36 (2024).

[70] H. LI, K. WU, C. ZHENG, Y. XIAO, H. WANG, Z. GENG, F. FENG, X. HE, AND P. WU, *Removing hidden confounding in recommendation: a unified multi-task learning approach*, in Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23, Red Hook, NY, USA, 2024, Curran Associates Inc.

[71] J. LI, B. HUI, G. QU, J. YANG, B. LI, B. LI, B. WANG, B. QIN, R. GENG, N. HUO, X. ZHOU, C. MA, G. LI, K. C. CHANG, F. HUANG, R. CHENG, AND Y. LI, *Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls*, in Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23, Red Hook, NY, USA, 2023, Curran Associates Inc.

[72] J. LI, Y. WANG, AND J. MCAULEY, *Time interval aware self-attention for sequential recommendation*, in Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM '20, New York, NY, USA, 2020, Association for Computing Machinery, p. 322–330.

[73] X. LI, Y. SUN, AND G. CHENG, *Tsqa: tabular scenario based question answering*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 13297–13305.

[74] X. LI, Y. ZHANG, AND E. C. MALTHOUSE, *Pbnr: Prompt-based news recommender system*, arXiv preprint arXiv:2304.07862, (2023).

[75] X. LI, R. ZHAO, Y. K. CHIA, B. DING, L. BING, S. JOTY, AND S. PORIA, *Chain of knowledge: A framework for grounding large language models with structured knowledge bases*, arXiv preprint arXiv:2305.13269, (2023).

[76] Y. LI, Y. HUANG, B. YANG, B. VENKITESH, A. LOCATELLI, H. YE, T. CAI, P. LEWIS, AND D. CHEN, *Snapkv: Llm knows what you are looking for before generation*, arXiv preprint arXiv:2404.14469, (2024).

[77] Y. Li, P. Wang, X. Zhu, A. Chen, H. Jiang, D. Cai, V. W. K. Chan, and J. Li, *GLBench: A comprehensive benchmark for graph with large language models*, in The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2024.

[78] Z. Li, Y. Du, M. Zheng, and M. Song, *MiMoTable: A multi-scale spreadsheet benchmark with meta operations for table reasoning*, in Proceedings of the 31st International Conference on Computational Linguistics, O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert, eds., Abu Dhabi, UAE, Jan. 2025, Association for Computational Linguistics, pp. 2548–2560.

[79] B. Lim, S. Ö. Arik, N. Loeff, and T. Pfister, *Temporal fusion transformers for interpretable multi-horizon time series forecasting*, International Journal of Forecasting, 37 (2021), pp. 1748–1764.

[80] LinkedIn, *Warehousing and storage market global comprehensive analysis and key players*.
`https://www.linkedin.com/pulse/warehousing-storage-market-global-comprehensiv`
May 2024.
[Online]. Available: https://www.linkedin.com/pulse/warehousing-storage-market-global-comprehensive-analysis-d7b9f/. [Accessed: 29 March 2025].

[81] A. Liu, X. Hu, L. Wen, and P. S. Yu, *A comprehensive evaluation of chatgpt's zero-shot text-to-sql capability*, arXiv preprint arXiv:2303.13547, (2023).

[82] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, *Lost in the middle: How language models use long contexts*, Transactions of the Association for Computational Linguistics, 12 (2024), pp. 157–173.

[83] P. Liu, Y. Ren, J. Tao, and Z. Ren, *Git-mol: A multi-modal large language model for molecular science with graph, image, and text*, Computers in biology and medicine, 171 (2024), p. 108073.

[84] Q. Liu, J. Zhu, Q. Dai, and X.-M. Wu, *Boosting deep ctr prediction with a plug-and-play pre-trainer for news recommendation*, in Proceedings of the 29th International Conference on Computational Linguistics, 2022, pp. 2823–2833.

[85] T. Liu, F. Wang, and M. Chen, *Rethinking tabular data understanding with large language models*, arXiv preprint arXiv:2312.16702, (2023).

[86] Z. Liu, K. Duan, J. Yang, H. Xu, M. Zhang, and S. Wang, *MetaFill: Text infilling for meta-path generation on heterogeneous information networks*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Y. Goldberg, Z. Kozareva, and Y. Zhang, eds., Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 5110–5122.

[87] P. Lu, B. Peng, H. Cheng, M. Galley, K.-W. Chang, Y. N. Wu, S.-C. Zhu, and J. Gao, *Chameleon: Plug-and-play compositional reasoning with large language models*, in Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[88] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, et al., *Self-refine: Iterative refinement with self-feedback*, Advances in Neural Information Processing Systems, 36 (2024).

[89] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, *Automating the construction of internet portals with machine learning*, Information Retrieval, 3 (2000), pp. 127–163.

[90] Meta, *Build the future of ai with meta Llama 3*, 2024.

[91] D. Min, N. Hu, R. Jin, N. Lin, J. Chen, Y. Chen, Y. Li, G. Qi, Y. Li, N. Li, et al., *Exploring the impact of table-to-text methods on augmenting llm-based question answering with domain hybrid data*, arXiv preprint arXiv:2402.12869, (2024).

[92] S. Min, D. Chen, L. Zettlemoyer, and H. Hajishirzi, *Knowledge guided text retrieval and reading for open domain question answering*, arXiv preprint arXiv:1911.03868, (2019).

[93] N. S. Moosavi, A. Rücklé, D. Roth, and I. Gurevych, *Scigen: a dataset for reasoning-aware text generation from scientific tables*, in Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021.

[94] L. Nan, C. Hsieh, Z. Mao, X. V. Lin, N. Verma, R. Zhang, W. Kryściński, H. Schoelkopf, R. Kong, X. Tang, M. Mutuma, B. Rosand, I. Trindade,

R. BANDARU, J. CUNNINGHAM, C. XIONG, D. RADEV, AND D. RADEV, *FeTaQA: Free-form table question answering*, Transactions of the Association for Computational Linguistics, 10 (2022), pp. 35–49.

[95] L. NAN, D. RADEV, R. ZHANG, A. RAU, A. SIVAPRASAD, C. HSIEH, X. TANG, A. VYAS, N. VERMA, P. KRISHNA, Y. LIU, N. IRWANTO, J. PAN, F. RAHMAN, A. ZAIDI, M. MUTUMA, Y. TARABAR, A. GUPTA, T. YU, Y. C. TAN, X. V. LIN, C. XIONG, R. SOCHER, AND N. F. RAJANI, *DART: Open-domain structured data record to text generation*, in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, eds., Online, June 2021, Association for Computational Linguistics, pp. 432–447.

[96] P. K. C. NIMMALAPUDI, *Mcrec: A co-attention based meta-path contextualized recommender system for heterogeneous information networks*, in 2025 International Conference on Computer, Electrical Communication Engineering (ICCECE), 2025, pp. 1–6.

[97] OPENAI, *Openai's gpt-3.5 turbo*, 2023.
Accessed: October 6, 2024.

[98] ——, *Gpt-4o mini: advancing cost-efficient intelligence*, 2024.

[99] A. PARIKH, X. WANG, S. GEHRMANN, M. FARUQUI, B. DHINGRA, D. YANG, AND D. DAS, *ToTTo: A controlled table-to-text generation dataset*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), B. Webber, T. Cohn, Y. He, and Y. Liu, eds., Online, Nov. 2020, Association for Computational Linguistics, pp. 1173–1186.

[100] P. PASUPAT AND P. LIANG, *Compositional semantic parsing on semi-structured tables*, in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 1470–1480.

[101] ——, *Compositional semantic parsing on semi-structured tables*, arXiv preprint arXiv:1508.00305, (2015).

[102] ——, *Compositional semantic parsing on semi-structured tables*, in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), C. Zong and M. Strube, eds., Beijing, China, July 2015, Association for Computational Linguistics, pp. 1470–1480.

[103] J. PEARL, *Causal inference in statistics: An overview*, (2009).

[104] P. PEZESHKPOUR AND E. HRUSCHKA, *Large language models sensitivity to the order of options in multiple-choice questions*, in Findings of the Association for Computational Linguistics: NAACL 2024, K. Duh, H. Gomez, and S. Bethard, eds., Mexico City, Mexico, June 2024, Association for Computational Linguistics, pp. 2006–2017.

[105] Q. QI, Z. DENG, Y. ZHU, L. J. LEE, M. WITBROCK, AND J. LIU, *TaKG: A new dataset for paragraph-level table-to-text generation enhanced with knowledge graphs*, in Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022, Y. He, H. Ji, S. Li, Y. Liu, and C.-H. Chang, eds., Online only, Nov. 2022, Association for Computational Linguistics, pp. 176–187.

[106] T. QI, F. WU, C. WU, AND Y. HUANG, *Pp-rec: News recommendation with personalized user interest and time-aware news popularity*, in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 5457–5467.

[107] ——, *News recommendation with candidate-aware user modeling*, in Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval, 2022, pp. 1917–1921.

[108] N. RAJKUMAR, R. LI, AND D. BAHDANAU, *Evaluating the text-to-sql capabilities of large language models*, arXiv preprint arXiv:2204.00498, (2022).

[109] S. RAZA AND C. DING, *News recommender system: a review of recent progress, challenges, and opportunities*, Artificial Intelligence Review, (2022), pp. 1–52.

[110] X. REN, J. TANG, D. YIN, N. CHAWLA, AND C. HUANG, *A survey of large language models for graphs*, in Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6616–6626.

[111] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, *Bpr: Bayesian personalized ranking from implicit feedback*, in Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Arlington, Virginia, USA, 2009, AUAI Press, p. 452–461.

[112] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, *Dynamic graph representation learning via self-attention networks*, arXiv preprint arXiv:1812.09430, (2018).

[113] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, *Modeling relational data with graph convolutional networks*, in The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15, Springer, 2018, pp. 593–607.

[114] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, *Collective classification in network data*, AI magazine, 29 (2008), pp. 93–93.

[115] Y. Shao and N. Nakashole, *On linearizing structured data in encoder-decoder language models: Insights from text-to-sql*, in Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 131–156.

[116] R. Shwartz-Ziv and A. Armon, *Tabular data: Deep learning is not all you need*, Inf. Fusion, 81 (2022), p. 84–90.

[117] A. Singha, J. Cambronero, S. Gulwani, V. Le, and C. Parnin, *Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms*, in NeurIPS 2023 Second Table Representation Learning Workshop.

[118] G. Somepalli, A. Schwarzschild, M. Goldblum, C. B. Bruss, and T. Goldstein, *Saint: Improved neural networks for tabular data via row attention and contrastive pre-training*, in NeurIPS 2022 First Table Representation Workshop.

[119] G. Sottocornola, P. Symeonidis, and M. Zanker, *Session-based news recommendations*, in Companion Proceedings of the The Web Conference 2018, 2018, pp. 1395–1399.

[120] Y. SUI, M. ZHOU, M. ZHOU, S. HAN, AND D. ZHANG, *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, WSDM '24, New York, NY, USA, 2024, Association for Computing Machinery, p. 645–654.

[121] F. SUN, J. LIU, J. WU, C. PEI, X. LIN, W. OU, AND P. JIANG, *Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer*, in Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, New York, NY, USA, 2019, Association for Computing Machinery, p. 1441–1450.

[122] X. TAN, H. WANG, X. QIU, Y. CHENG, Y. XU, W. CHU, AND Y. QI, *Struct-x: Enhancing large language models reasoning with structured data*, arXiv preprint arXiv:2407.12522, (2024).

[123] Z. TAN, X. LIU, Q. SHU, X. LI, C. WAN, D. LIU, Q. WAN, AND G. LIAO, *Enhancing text-to-SQL capabilities of large language models through tailored promptings*, in Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, eds., Torino, Italia, May 2024, ELRA and ICCL, pp. 6091–6109.

[124] W. C. TCHUITCHEU, T. LU, AND A. DOOMS, *Table representation learning using heterogeneous graph embedding*, Pattern Recogn., 156 (2024).

[125] G. TEAM, T. MESNARD, C. HARDIN, R. DADASHI, S. BHUPATIRAJU, S. PATHAK, L. SIFRE, M. RIVIÈRE, M. S. KALE, J. LOVE, P. TAFTI, L. HUSSENOT, P. G. SESSA, A. CHOWDHERY, A. ROBERTS, A. BARUA, A. BOTEV, A. CASTRO-ROS, A. SLONE, A. HÉLIOU, A. TACCHETTI, A. BULANOVA, A. PATERSON, B. TSAI, B. SHAHRIARI, C. L. LAN, C. A. CHOQUETTE-CHOO, C. CREPY, D. CER, D. IPPOLITO, D. REID, E. BUCHATSKAYA, E. NI, E. NOLAND, G. YAN, G. TUCKER, G.-C. MURARU, G. ROZHDESTVENSKIY, H. MICHALEWSKI, I. TENNEY, I. GRISHCHENKO, J. AUSTIN, J. KEELING, J. LABANOWSKI, J.-B. LESPIAU, J. STANWAY, J. BRENNAN, J. CHEN, J. FERRET, J. CHIU, J. MAO-JONES, K. LEE, K. YU, K. MILLICAN, L. L. SJOESUND, L. LEE, L. DIXON, M. REID, M. MIKUŁA, M. WIRTH, M. SHARMAN, N. CHINAEV, N. THAIN, O. BACHEM, O. CHANG, O. WAHLTINEZ, P. BAILEY, P. MICHEL,

P. Yotov, R. Chaabouni, R. Comanescu, R. Jana, R. Anil, R. McIlroy, R. Liu, R. Mullins, S. L. Smith, S. Borgeaud, S. Girgin, S. Douglas, S. Pandya, S. Shakeri, S. De, T. Klimenko, T. Hennigan, V. Feinberg, W. Stokowiec, Y. hui Chen, Z. Ahmed, Z. Gong, T. Warkentin, L. Peran, M. Giang, C. Farabet, O. Vinyals, J. Dean, K. Kavukcuoglu, D. Hassabis, Z. Ghahramani, D. Eck, J. Barral, F. Pereira, E. Collins, A. Joulin, N. Fiedel, E. Senter, A. Andreev, and K. Kenealy, *Gemma: Open models based on gemini research and technology*, 2024.

[126] Y. Tian, H. Song, Z. Wang, H. Wang, Z. Hu, F. Wang, N. V. Chawla, and P. Xu, *Graph neural prompting with large language models*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, 2024, pp. 19080–19088.

[127] H. Touvron et al., *The llama 3 herd of models*, 2024.

[128] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., *Llama: Open and efficient foundation language models*, arXiv preprint arXiv:2302.13971, (2023).

[129] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, *Llama: Open and efficient foundation language models*, 2023.

[130] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., *Llama 2: Open foundation and fine-tuned chat models*, arXiv preprint arXiv:2307.09288, (2023).

[131] M. Trevisiol, L. M. Aiello, R. Schifanella, and A. Jaimes, *Cold-start news recommendation with domain-dependent browse graph*, in Proceedings of the 8th ACM Conference on Recommender systems, 2014, pp. 81–88.

[132] S. Tripathi, B. A. Fritz, M. Abdelhack, M. S. Avidan, Y. Chen, and C. R. King, *Multi-view representation learning for tabular data integration using inter-feature relationships*, Journal of Biomedical Informatics, 151 (2024), p. 104602.

[133] H. TRIVEDI, N. BALASUBRAMANIAN, T. KHOT, AND A. SABHARWAL, *MuSiQue: Multihop Questions via Single-hop Question Composition*, Transactions of the Association for Computational Linguistics, 10 (2022), pp. 539–554.

[134] A. TVERSKY AND D. KAHNEMAN, *The framing of decisions and the psychology of choice*, science, 211 (1981), pp. 453–458.

[135] T. UCAR, E. HAJIRAMEZANALI, AND L. EDWARDS, *Subtab: Subsetting features of tabular data for self-supervised representation learning*, Advances in Neural Information Processing Systems, 34 (2021), pp. 18853–18865.

[136] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, Advances in neural information processing systems, 30 (2017).

[137] VICUNA, *Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality*, 2023.

[138] A. VLONTZOS, B. KAINZ, AND C. M. GILLIGAN-LEE, *Estimating categorical counterfactuals via deep twin networks*, Nature Machine Intelligence, 5 (2023), pp. 159–168.

[139] H. WANG, K. HU, H. DONG, AND L. GAO, *Doctabqa: Answering questions from long documents using tables*, in International Conference on Document Analysis and Recognition, Springer, 2024, pp. 470–487.

[140] H. WANG, F. ZHANG, X. XIE, AND M. GUO, *Dkn: Deep knowledge-aware network for news recommendation*, in Proceedings of the 2018 world wide web conference, 2018, pp. 1835–1844.

[141] J. WANG, Y. CHEN, Z. WANG, AND W. ZHAO, *Popularity-enhanced news recommendation with multi-view interest representation*, in Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21, New York, NY, USA, 2021, Association for Computing Machinery, p. 1949–1958.

[142] N. X. R. WANG, D. MAHAJAN, M. DANILEVSKY, AND S. ROSENTHAL, *SemEval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (SEM-TAB-FACTS)*, in Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), A. Palmer, N. Schneider,

N. Schluter, G. Emerson, A. Herbelot, and X. Zhu, eds., Online, Aug. 2021, Association for Computational Linguistics, pp. 317–326.

[143] R. Wang, S. Wang, W. Lu, and X. Peng, *News recommendation via multi-interest news sequence modelling*, in ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 7942–7946.

[144] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, *Neural graph collaborative filtering*, in Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19, New York, NY, USA, 2019, Association for Computing Machinery, p. 165–174.

[145] X. Wang, Q. Li, D. Yu, Z. Wang, H. Chen, and G. Xu, *Mgpolicy: Meta graph enhanced off-policy learning for recommendations*, in Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22, New York, NY, USA, 2022, Association for Computing Machinery, p. 1369–1378.

[146] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou, *Self-consistency improves chain of thought reasoning in language models*, in The Eleventh International Conference on Learning Representations, 2023.

[147] Z. Wang, H. Zhang, C.-L. Li, J. M. Eisenschlos, V. Perot, Z. Wang, L. Miculicich, Y. Fujii, J. Shang, C.-Y. Lee, et al., *Chain-of-table: Evolving tables in the reasoning chain for table understanding*, in The Twelfth International Conference on Learning Representations, 2024.

[148] Z. Wang, H. Zhang, C.-L. Li, J. M. Eisenschlos, V. Perot, Z. Wang, L. Miculicich, Y. Fujii, J. Shang, C.-Y. Lee, and T. Pfister, *Chain-of-table: Evolving tables in the reasoning chain for table understanding*, in The Twelfth International Conference on Learning Representations, 2024.

[149] Z. Wang, J. Zhang, H. Xu, X. Chen, Y. Zhang, W. X. Zhao, and J.-R. Wen, *Counterfactual data-augmented sequential recommendation*, in Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, New York, NY, USA, 2021, Association for Computing Machinery, p. 347–356.

[150] J. WEI, X. WANG, D. SCHUURMANS, M. BOSMA, F. XIA, E. CHI, Q. V. LE, D. ZHOU, ET AL., *Chain-of-thought prompting elicits reasoning in large language models*, Advances in neural information processing systems, 35 (2022), pp. 24824–24837.

[151] E. A. WENHU CHEN, *Tabfact : A large-scale dataset for table-based fact verification*, in International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, April 2020.

[152] WIKIPEDIA, *Electronic health records in the united states*. `https://en.wikipedia.org/wiki/Electronic_health_records_in_the_United_States#Start-up_costs`, Mar. 2025.

[153] C. WU, F. WU, M. AN, J. HUANG, Y. HUANG, AND X. XIE, *Neural news recommendation with attentive multi-view learning*, arXiv preprint arXiv:1907.05576, (2019).

[154] ——, *Npa: neural news recommendation with personalized attention*, in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2576–2584.

[155] C. WU, F. WU, S. GE, T. QI, Y. HUANG, AND X. XIE, *Neural news recommendation with multi-head self-attention*, 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), 2019, pp. 6389–6394.

[156] F. WU, Y. QIAO, J.-H. CHEN, C. WU, T. QI, J. LIAN, D. LIU, X. XIE, J. GAO, W. WU, AND M. ZHOU, *MIND: A large-scale dataset for news recommendation*, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds., Online, July 2020, Association for Computational Linguistics, pp. 3597–3606.

[157] X. WU, J. YANG, L. CHAI, G. ZHANG, J. LIU, X. DU, D. LIANG, D. SHU, X. CHENG, T. SUN, ET AL., *Tablebench: A comprehensive and complex benchmark for table question answering*, arXiv preprint arXiv:2408.09174, (2024).

[158] L. XIA, C. HUANG, Y. XU, P. DAI, M. LU, AND L. BO, *Multi-behavior enhanced recommendation with cross-interaction collaborative relation modeling*, in 2021

IEEE 37th International Conference on Data Engineering (ICDE), IEEE, 2021, pp. 1931–1936.

[159] L. Xia, C. Huang, Y. Xu, P. Dai, B. Zhang, and L. Bo, *Multiplex behavioral relation learning for recommendation via memory augmented transformer network*, in Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, 2020, pp. 2397–2406.

[160] L. Xia, Y. Xu, C. Huang, P. Dai, and L. Bo, *Graph meta network for multi-behavior recommendation*, in Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, 2021, pp. 757–766.

[161] S. Xu, J. Tan, S. Heinecke, V. J. Li, and Y. Zhang, *Deconfounded causal collaborative filtering*, ACM Transactions on Recommender Systems, (2021).

[162] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, *Hypergcn: A new method for training graph convolutional networks on hypergraphs*, Advances in neural information processing systems, 32 (2019).

[163] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, and Z. Fan, *Qwen2 technical report*, arXiv preprint arXiv:2407.10671, (2024).

[164] H. Yang, H. Chen, L. Li, S. Y. Philip, and G. Xu, *Hyper meta-path contrastive learning for multi-behavior recommendation*, in 2021 IEEE International Conference on Data Mining (ICDM), IEEE, 2021, pp. 787–796.

[165] H. Yang, H. Chen, S. Zhang, X. Sun, Q. Li, X. Zhao, and G. Xu, *Generating counterfactual hard negative samples for graph contrastive learning*, in Proceedings of the ACM Web Conference 2023, WWW '23, New York, NY, USA, 2023, Association for Computing Machinery, p. 621–629.

[166] Q. YANG, Q. CHEN, W. WANG, B. HU, AND M. ZHANG, *Enhancing multi-modal multi-hop question answering via structured knowledge and unified retrieval-generation*, in Proceedings of the 31st ACM International Conference on Multimedia, MM '23, New York, NY, USA, 2023, Association for Computing Machinery, p. 5223–5234.

[167] Y. YANG, H. YIN, J. CAO, T. CHEN, Q. V. H. NGUYEN, X. ZHOU, AND L. CHEN, *Time-aware dynamic graph embedding for asynchronous structural evolution*, IEEE Transactions on Knowledge and Data Engineering, 35 (2023), pp. 9656–9670.

[168] Z. YANG, P. QI, S. ZHANG, Y. BENGIO, W. COHEN, R. SALAKHUTDINOV, AND C. D. MANNING, *HotpotQA: A dataset for diverse, explainable multi-hop question answering*, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, eds., Brussels, Belgium, Oct.-Nov. 2018, Association for Computational Linguistics, pp. 2369–2380.

[169] Y. YE, B. HUI, M. YANG, B. LI, F. HUANG, AND Y. LI, *Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning*, in Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, New York, NY, USA, 2023, Association for Computing Machinery, p. 174–184.

[170] ——, *Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning*, in Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, New York, NY, USA, 2023, Association for Computing Machinery, p. 174–184.

[171] ——, *Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning*, in Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, New York, NY, USA, 2023, Association for Computing Machinery, p. 174–184.

[172] P. YIN, W.-D. LI, K. XIAO, A. RAO, Y. WEN, K. SHI, J. HOWLAND, P. BAILEY, M. CATASTA, H. MICHALEWSKI, O. POLOZOV, AND C. SUTTON, *Natural*

language to code generation in interactive data science notebooks, in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), A. Rogers, J. Boyd-Graber, and N. Okazaki, eds., Toronto, Canada, July 2023, Association for Computational Linguistics, pp. 126–173.

[173] P. YIN, G. NEUBIG, W.-T. YIH, AND S. RIEDEL, *Tabert: Pretraining for joint understanding of textual and tabular data*, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8413–8426.

[174] T. YU, R. ZHANG, K. YANG, M. YASUNAGA, D. WANG, Z. LI, J. MA, I. LI, Q. YAO, S. ROMAN, Z. ZHANG, AND D. RADEV, *Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task*, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, eds., Brussels, Belgium, Oct.-Nov. 2018, Association for Computational Linguistics, pp. 3911–3921.

[175] E. YUAN, W. GUO, Z. HE, H. GUO, C. LIU, AND R. TANG, *Multi-behavior sequential transformer recommender*, in Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22, New York, NY, USA, 2022, Association for Computing Machinery, p. 1642–1652.

[176] G. ZABËRGJA, A. KADRA, AND J. GRABOCKA, *Tabular data: Is attention all you need?*, arXiv preprint arXiv:2402.03970, (2024).

[177] Q. ZHANG, X. ZHANG, Y. LIU, H. WANG, M. GAO, J. ZHANG, AND R. GUO, *Debiasing recommendation by learning identifiable latent confounders*, in Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23, New York, NY, USA, 2023, Association for Computing Machinery, p. 3353–3363.

[178] T. ZHANG, X. YUE, Y. LI, AND H. SUN, *Tablellama: Towards open large generalist models for tables*, in Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 6024–6044.

[179] Y. ZHANG, F. FENG, J. ZHANG, K. BAO, Q. WANG, AND X. HE, *Collm: Integrating collaborative embeddings into large language models for recommendation*, IEEE Transactions on Knowledge and Data Engineering, (2025).

[180] Y. ZHANG, X. SUI, F. PAN, K. YU, K. LI, S. TIAN, A. ERDENGASILENG, Q. HAN, W. WANG, J. WANG, ET AL., *A comprehensive large-scale biomedical knowledge graph for ai-powered data-driven biomedical research*, Nature Machine Intelligence, (2025), pp. 1–13.

[181] Z. ZHANG, Y. SHENG, T. ZHOU, T. CHEN, L. ZHENG, R. CAI, Z. SONG, Y. TIAN, C. RE, C. BARRETT, Z. WANG, AND B. CHEN, *H2o: Heavy-hitter oracle for efficient generative inference of large language models*, in Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[182] Y. ZHAO, H. ZHANG, S. SI, L. NAN, X. TANG, AND A. COHAN, *Investigating table-to-text generation capabilities of llms in real-world information seeking scenarios*, arXiv preprint arXiv:2305.14987, (2023).

[183] Y. ZHAO, C. ZHAO, L. NAN, Z. QI, W. ZHANG, X. TANG, B. MI, AND D. RADEV, *Robut: A systematic study of table qa robustness against human-annotated adversarial perturbations*, in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 6064–6081.

[184] C. ZHENG, H. ZHOU, F. MENG, J. ZHOU, AND M. HUANG, *Large language models are not robust multiple choice selectors*, in The Twelfth International Conference on Learning Representations, 2024.

[185] L. ZHENG, W.-L. CHIANG, Y. SHENG, S. ZHUANG, Z. WU, Y. ZHUANG, Z. LIN, Z. LI, D. LI, E. P. XING, H. ZHANG, J. E. GONZALEZ, AND I. STOICA, *Judging llm-as-a-judge with mt-bench and chatbot arena*, in Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23, Red Hook, NY, USA, 2024, Curran Associates Inc.

[186] V. ZHONG, C. XIONG, AND R. SOCHER, *Seq2SQL: Generating structured queries from natural language using reinforcement learning*, 2018.

[187] W. ZHOU, M. MESGAR, H. ADEL, AND A. FRIEDRICH, *FREB-TQA: A fine-grained robustness evaluation benchmark for table question answering*, in Proceedings

of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), K. Duh, H. Gomez, and S. Bethard, eds., Mexico City, Mexico, June 2024, Association for Computational Linguistics, pp. 2479–2497.

[188] F. Zhu, W. Lei, Y. Huang, C. Wang, S. Zhang, J. Lv, F. Feng, and T.-S. Chua, *TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance*, in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), C. Zong, F. Xia, W. Li, and R. Navigli, eds., Online, Aug. 2021, Association for Computational Linguistics, pp. 3277–3287.

[189] M. Zhu, S. Stanivuk, A. Petrovic, M. Nikolic, and P. Lio, *Incorporating llm priors into tabular learners*, in NeurIPS 2023 Second Table Representation Learning Workshop.

[190] X. Zhu, Y. Zhang, F. Feng, X. Yang, D. Wang, and X. He, *Mitigating hidden confounding effects for causal recommendation*, arXiv preprint arXiv:2205.07499, (2022).