
AI translation · View original & related papers at
chinaxiv.org/items/chinaxiv-202601.00187

Construct, Align, and Reason: Large Ontology Models for Enterprise Knowledge Management

Authors: Yao Zhang, Could you clarify what you'd like me to do with "Hongyin Zhu" ?

For example, do you want:

- Information about this person (e.g., an academic profile or research area), or
- Their name rendered in a specific format (e.g., citation style, transliteration), or
- Something translated that is related to this author?, Hongyin

Date: 2026-01-20T20:18:30+00:00

Abstract

Enterprise-scale knowledge management faces substantial challenges in integrating multi-source heterogeneous data and enabling effective semantic reasoning. Traditional knowledge graphs often struggle to uncover implicit relationships and lack sufficient semantic understanding to support complex question answering.

To address these limitations, we propose a unified construct-align-reason framework, the Large Ontology Model (LOM). We first construct a dual-layer enterprise ontology from structured databases and unstructured text, and then fuse these sources into a comprehensive enterprise ontology.

To enable instruction-aligned reasoning, we design a unified three-stage training pipeline: (1) ontology instruction fine-tuning to enhance structural understanding; (2) text-ontology grounding to strengthen semantic encoding of nodes; and (3) multi-task instruction tuning on ontology-language pairs with curriculum learning to improve semantic reasoning and generation capabilities.

We further build comprehensive training and evaluation datasets that cover a wide range of ontology reasoning tasks. On this benchmark, our 4B-parameter LOM achieves 89.47% accuracy and outperforms DeepSeek-V3.2 on complex

graph reasoning, demonstrating effective fusion of ontology structure and natural language.

Full Text

Preamble

Construct, Align, and Reason: Large Ontology Models for Enterprise Knowledge Management

Yao Zhanga,b, Hongyin Zhua,b

aYonyou AI Lab, bYonyou Network Technology Co., Ltd.

Keywords: ontology construction, large language model, knowledge graph, graph encoder

Enterprise-scale knowledge management faces significant challenges in integrating multi-source heterogeneous data and enabling effective semantic reasoning. Traditional knowledge graphs often struggle with implicit relationship discovery and lack sufficient semantic understanding for complex question answering. To address these limitations, we introduce a unified construct-align-reason framework, the large ontology model (LOM). We first build a dual-layer enterprise ontology from structured databases and unstructured text, subsequently fusing these sources into a comprehensive enterprise ontology. To enable instruction-aligned reasoning, we propose a unified three-stage training pipeline: ontology instruction fine-tuning to improve structural understanding; text-ontology grounding to strengthen node semantic encoding; and multi-task instruction tuning on ontology-language pairs with curriculum learning to enhance semantic reasoning and generation. We also construct comprehensive training and evaluation datasets covering diverse ontology reasoning tasks. On this benchmark, our 4B-parameter LOM achieves 89.47% accuracy and outperforms DeepSeek-V3.2 on complex graph reasoning, indicating effective fusion of ontology structure and language.

1. Introduction

In the wave of digital transformation, enterprise knowledge management faces unprecedented challenges. Traditional systems typically rely on relational databases or basic document management systems to store and manage enterprise knowledge. While these approaches can store structured data effectively, they struggle with complex unstructured knowledge and inter-entity relationships. In recent years, knowledge graphs have emerged as a powerful representation that models enterprise knowledge as graphs, using nodes and edges to capture entities and their relationships, thereby enabling more intelligent querying and reasoning.

Existing approaches for enterprise ontology construction and reasoning face distinct limitations. On the construction side, methods often treat structured

databases and unstructured text in isolation. Traditional schema mapping tools [?] struggle to identify implicit relationships (e.g., missing foreign keys) in legacy databases, while standard information extraction models [?] lack the domain adaptability to merge these structured backbones with rich textual knowledge.

On the reasoning side, a significant semantic gap persists: graph neural networks (GNNs) [?] capture topology but lack the reasoning depth for complex business questions, whereas large language models (LLMs) [?] possess semantic knowledge but often fail to adhere to rigorous graph structures. This disconnect hampers the ability to perform reliable, multi-hop reasoning over heterogeneous enterprise data. To address these challenges, we present a large ontology model (LOM) that unifies multi-source ontology construction with a structure-aware instruction tuning pipeline, effectively bridging the divide between data integration and semantic reasoning.

Constructing an enterprise ontology is a non-trivial undertaking. We employ a layered approach to build ontologies from both structured databases and unstructured text. For structured databases, where explicit foreign keys are frequently missing, we propose a multi-factor relationship discovery algorithm that analyzes both schema metadata and data content overlap to uncover implicit connections. This enables the construction of a dual-layer ontology comprising an abstract schema layer and a concrete instance layer. For unstructured text, we utilize an LLM&LOM-based pipeline that performs entity-relation extraction, link prediction, and robust entity disambiguation via a hybrid of symbolic rules and semantic embeddings. Finally, we fuse these heterogeneous sources into a unified enterprise ontology through cross-modal alignment based on tag-description matching.

The second challenge lies in training an LOM capable of deeply understanding and reasoning over these heterogeneous enterprise ontologies. Existing methods often fail to bridge the semantic gap between graph structures and textual knowledge. To address this, we implement a unified three-stage training pipeline. First, we employ ontology instruction fine-tuning to endow the LLM with foundational ontology structural understanding. Second, we introduce a text-ontology grounding stage that trains an alignment projector to fuse textual semantics with ontology features via intra- and inter-type alignment. Finally, we conduct multi-task instruction tuning over ontology-language pairs with curriculum learning, guiding the model from simple predictive tasks to complex generative reasoning. To support this pipeline, we construct a comprehensive CoT-enhanced dataset that captures algorithmic reasoning paths, enabling the model to learn the logic behind ontology-centric operations rather than simple answer mapping.

We conduct systematic evaluation on our datasets, and the 4B-parameter LOM achieves state-of-the-art performance. Our main contributions are:

1. We introduce the large ontology model, a unified construct-align-reason framework that systematically integrates multi-source ontology construc-

tion, text-ontology alignment, and instruction-aligned reasoning to resolve the semantic disconnect between structured databases and unstructured knowledge.

2. We design a three-stage instruction alignment training pipeline—comprising ontology instruction fine-tuning, text-ontology grounding, and multi-task instruction tuning over ontology-language pairs—to endow the model with robust complex ontology reasoning capabilities.
3. We construct a comprehensive CoT-enhanced dataset covering 19 diverse graph reasoning tasks. Extensive evaluations demonstrate that our LOM-4B achieves state-of-the-art performance with 89.47% accuracy on this benchmark, significantly outperforming leading baselines in complex graph reasoning tasks.

2.1. Ontology Construction from Heterogeneous

Recent studies have revisited ontology construction from relational databases by incorporating agent-based reasoning and large language models to reduce manual schema engineering. Trajanoska et al. [?] propose a multi-agent framework in which specialized agents collaboratively perform schema interpretation, mapping rule generation, and semantic alignment between relational tables and knowledge graph schemas. While this approach improves modularity and interpretability, it still relies on explicit coordination protocols and assumes relatively clean schema-level signals, limiting its robustness in large-scale enterprise databases with implicit or noisy relational structures.

More recently, retrieval-augmented and autonomous ontology construction methods have emerged. Nayyeri et al. [?] introduce a RAG-based framework that retrieves schema fragments and instance-level evidence to guide LLM-driven ontology generation, demonstrating improved adaptability across heterogeneous databases. In parallel, AutoSchemaKG [?] explores dynamic schema induction from web-scale corpora, enabling autonomous ontology evolution without predefined schemas. However, these methods primarily focus on either structured databases or unstructured text in isolation, and they do not explicitly address cross-source ontology alignment or the discovery of implicit relational patterns such as hidden foreign keys and enterprise-specific relationship semantics.

A complementary line of research focuses on declarative mapping frameworks for constructing RDF knowledge graphs from structured and heterogeneous data sources. SDM-RDFizer [?] presents an efficient RML interpreter optimized for large-scale RDF generation through parallel execution and standards-compliant mapping, enabling deterministic and reproducible knowledge graph construction. Building upon such infrastructures, Assche et al. [?] study backward compatibility and rule reuse in RML mapping pipelines, addressing maintainability challenges under evolving schemas. While these systems provide robust engineering solutions for knowledge graph materialization, they rely on manually

defined mappings and predefined schemas, and do not address automated ontology induction, implicit relationship discovery, or instruction-aligned reasoning over enterprise knowledge graphs, which are the focus of our work.

2.2. Ontology Reasoning

Recent advances in large language models have motivated the integration of graph-structured knowledge into language-centric reasoning frameworks. Early graph language models focus on encoding graph structures into LLM-compatible representations. HiGPT [?] introduces a heterogeneous graph language model that aligns graph nodes and relations with textual embeddings, enabling joint reasoning over graph topology and semantics. Building upon this paradigm, GraphAgent [?] frames graph reasoning as an agentic process, allowing LLMs to iteratively plan, reason, and interact with graph environments. These approaches demonstrate the potential of agent-based graph reasoning, yet they primarily target generic graph benchmarks and do not explicitly consider the scale, heterogeneity, and dynamic evolution characteristic of enterprise ontologies.

To enhance LLMs' capability to reason over graph structures, recent work explores instruction tuning and foundation model designs tailored to graph data. GraphInstruct [?] proposes a large-scale instruction dataset covering diverse graph algorithms and demonstrates that instruction-tuned LLMs can acquire non-trivial graph reasoning abilities without explicit graph neural networks. Complementarily, G-Reasoner [?] presents a foundation model for unified reasoning over graph-structured knowledge, emphasizing generalization across tasks and graph modalities. While these methods significantly advance instruction-aligned graph reasoning, they mainly focus on static or synthetic graph settings and provide limited support for grounding graph reasoning in real-world enterprise semantics and heterogeneous knowledge sources.

Another line of research enhances LLM reasoning through explicit interaction with knowledge graphs [?]. Survey work [?] systematically reviews the emerging capabilities of LLMs in KG construction and reasoning, highlighting both opportunities and limitations. On the modeling side, retrieval- and path-based approaches aim to improve multi-hop reasoning. GNN-RAG [?] employs graph neural networks to guide retrieval of relevant subgraphs for LLM reasoning, while Zhou et al. [?] introduce reflective mechanisms to iteratively refine KG-based reasoning. Paths-over-Graph [?] further explores path-centric reasoning strategies for complex multi-hop queries. In applied settings, Mendes et al. [?] demonstrate the feasibility of deploying LLM-powered KGQA systems in enterprise environments. However, these approaches typically treat the knowledge graph as an external tool or retrieval source, rather than jointly modeling ontology construction, graph representation, and instruction-aligned reasoning within a unified framework.

3. Approach

In this section, we present the LOM-based construct-align-reason framework, where construct denotes ontology construction, align denotes ontology-text alignment, and reason denotes generative ontology reasoning. We first introduce the model architecture, followed by unified training and inference strategies for complex ontology reasoning. We then detail the construction of our CoT-enhanced graph reasoning dataset. Finally, we describe our ontology construction methodology, which integrates structured databases and unstructured text into a unified enterprise ontology.

3.1. Large Ontology Model

3.1.1. Model Architecture **Figure 1 [Figure 1: see original paper]: Model Architecture Overview**

As illustrated in Figure 1, the LOM architecture is situated within the enterprise knowledge operations platform, bridging bottom-layer business systems & data (structured and unstructured data) with top-layer ontology applications. Specifically, an ontology construction layer is introduced, constructed from structured and unstructured data. Data flows through an ontology interface into the model, which employs a heterogeneous graph representation $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{N}, \mathcal{R})$ to jointly model structured and unstructured data. The core processing unit consists of three key components: (1) a graph encoder [?] that utilizes a graph transformer to capture structural dependencies; (2) a user input module that processes natural language inquiries (e.g., procurement requests) via a text encoder; and (3) the LOM hybrid cluster, where a linear alignment projector maps graph features into the LOM's embedding space. This deep fusion enables the model to perform CoT reasoning over enterprise ontologies.

3.1.2. Training Method We begin by fine-tuning the LLM using instruction tuning [?] to enhance ontology-centric understanding and reasoning, and then integrate it into the large ontology model framework. The training objective is:

$$\mathcal{L} = -\mathbb{E}_{\mathcal{G}, \mathbf{q}, \mathbf{a} \sim \mathcal{D}} \log P(\mathbf{a} | (\mathcal{G}, \mathbf{q}); \theta)$$

where \mathcal{G} is the graph, \mathbf{q} the query, \mathbf{a} the answer, and θ the model parameters. Given $(\mathcal{G}, \mathbf{q})$, this model is trained under θ to output correct \mathbf{a} , endowing the model with foundational graph understanding and reasoning.

During the training process, we align not only the knowledge graph data but also the ontology structure. Then we train the alignment projector and graph-token embeddings to align features between the text encoder and GNN. Specifically, we employ graph instruction alignment, where the LLM is trained to understand graph-structured data via graph-token-instruction pairs. We define two alignment datasets for this stage:

Intra-type alignment enhances understanding of tokens within a single meta-type by training the LLM to output the correct text sequence for a given graph-token sequence. The dataset is defined as:

$$\mathcal{D}_{\text{intra}} = \{[(\mathbf{e}_k, s_i), \dots], [(\mathbf{c}_k, \mathbf{c}_{s_i}), \dots]\}$$

We optimize this alignment using a next-token-prediction cross-entropy objective:

$$\mathcal{L}_{\text{intra}} = \mathbb{E}_{d \sim \mathcal{D}_{\text{intra}}} [\text{CE}(d[0] \mid \text{LLM}(d[1]))]$$

Inter-type alignment introduces multiple meta-types for complex heterogeneous relations, using tokens from different meta-types:

$$\mathcal{D}_{\text{inter}} = \{[(\mathbf{e}_m, s_m), (\mathbf{e}_n, s_n), \dots], [(\mathbf{c}_m, \mathbf{c}_{s_m}), (\mathbf{c}_n, \mathbf{c}_{s_n}), \dots]\}$$

Similarly, we train the LLM to predict the text sequence from the heterogeneous graph tokens:

$$\mathcal{L}_{\text{inter}} = \mathbb{E}_{d \sim \mathcal{D}_{\text{inter}}} [\text{CE}(d[0] \mid \text{LLM}(d[1]))]$$

where $\text{CE}(\cdot)$ denotes the cross-entropy loss function. In both intra- and inter-type alignment tasks, $d[0]$ represents the target text sequence (ground truth), while $d[1]$ is the input sequence of graph tokens processed by the LLM. The graph token definitions are: \mathbf{e}_i is the i -th graph token, s_i its meta-type, and \mathbf{c}_i the corresponding text description. This builds a foundation for understanding graph-structured data.

Finally, we perform multi-task fine-tuning based on the above alignment to enhance the model's capabilities across predictive and generative tasks. This phase utilizes diverse ontology-language instruction formats defined as:

$$\mathcal{D}_{\text{multi}} = \{ \{(\mathbf{x}_{\text{pred}}, \mathbf{x}_{\text{reasoning}}) \mid \mathbf{x}_{\text{gen}}\}, \{\mathcal{G}_{\text{exp}} \mid \mathcal{G}_{\text{skg}}\}, \mathbf{t}_i, \mathbf{a}_i \}$$

The model is optimized to generate the target answer \mathbf{a}_i given the instruction \mathbf{t}_i and graph context \mathcal{G} :

$$\mathcal{L}_{\text{multi}} = -\mathbb{E}_{(\mathbf{t}, \mathbf{a}, \mathcal{G}) \sim \mathcal{D}_{\text{multi}}} \log P(\mathbf{a} \mid \mathbf{t}, \mathcal{G}; \theta)$$

where $P(\mathbf{a} \mid \mathbf{t}, \mathcal{G}; \theta)$ represents the conditional probability of generating the correct answer \mathbf{a} given the instruction \mathbf{t} and graph context \mathcal{G} , parameterized by the model weights θ . The input features \mathbf{x}_{pred} , $\mathbf{x}_{\text{reasoning}}$, and \mathbf{x}_{gen} correspond to predictive, reasoning, and generative tasks, respectively. \mathcal{G}_{exp} denotes

explicit graph topology, while \mathcal{G}_{skg} refers to the schema-enhanced knowledge graph structure. The pair $(\mathbf{t}_i, \mathbf{a}_i)$ corresponds to the task-specific instruction and the ground-truth answer.

To ensure stable convergence and effective knowledge acquisition, we adopt a curriculum learning strategy that organizes training samples by difficulty. The model is first exposed to simpler predictive tasks before progressing to complex generative reasoning scenarios, thereby systematically building its capability to handle diverse graph-instruction pairs.

3.1.3. Dataset Construction Existing graph-task datasets primarily enable LLMs to learn simple reasoning tasks, while complex graph reasoning tasks (e.g., minimum spanning tree, PageRank) remain difficult to learn. To address this, we introduce a CoT-enhanced dataset for learning graph reasoning. The dataset includes problem descriptions, chain-of-thought reasoning, and final answers, enabling LLMs to learn the reasoning processes of complex reasoning tasks and improve performance on graph tasks.

We further design trajectory-generation problems for reasoning tasks via a CoT-enhanced trace generator implementing four core reasoning tasks: Dijkstra shortest path, Kahn topological sorting, Prim minimum spanning tree, and predecessor node search. A stepwise recording system captures intermediate states during execution, including initialization, node visitation, and state updates, and produces natural language chain-of-thought explanations. This converts complex reasoning execution into structured training data, allowing LLMs to learn full reasoning logic rather than memorizing answers.

The final dataset contains 115k training samples spanning 19 graph reasoning tasks, from simple traversals (BFS, DFS) to complex reasoning tasks (shortest path, MST, PageRank). Data are split into two stages: 20k samples for foundational graph language model training and 95k samples for deep fusion training. We additionally build 190 evaluation samples (10 per task) for performance assessment. All data are stored in PyG format, including node feature encodings (sentence-transformer, 768-dim) and graph structure, supporting both heterogeneous and homogeneous graph training.

3.2. Ontology Construction

3.2.1. Ontology Construction from Structured Data To address the semantic impedance mismatch and scalability challenges inherent in converting heterogeneous relational databases into ontologies, we propose a layered, decoupled, and incrementally evolving construction framework. We adopt a two-stage scanning strategy to build the ontology $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S)$. First, we extract schemas from the source tables $\mathcal{T} = \{T_1, \dots, T_n\}$ to identify explicit keys (e.g., fields containing ‘id’ or ‘code’). To complement this, we utilize LOMs to detect implicit associations, capturing semantic variants like ‘number’ that elude simple keyword matching. Second, we perform a sliding-window deep scan to

infer implicit foreign keys, utilizing a random sample of 1,000 records to assess value overlap. We define a multi-factor confidence function $\Phi(\cdot)$ to quantify the relationship between columns $c \in T_i$ and $c' \in T_j$:

$$\Phi(c, c') = \lambda_1 \cdot \text{name}(c, c') + \lambda_2 \cdot \mathbb{1}_{\text{type}}(c, c') + \lambda_3 \cdot \text{overlap}(c, c') + \lambda_4 \cdot \text{card}(c, c')$$

where name measures semantic similarity, $\mathbb{1}_{\text{type}}$ ensures data type compatibility, overlap quantifies value distribution overlap, and card infers cardinality patterns (e.g., one-to-many). An edge is established if $\Phi > \delta$. We construct a dual-layer ontology: the inter-table layer forms a graph via the above method, refined by k -core decomposition to retain the topological backbone, while tables are categorized top-down to abstract higher-level concepts. Finally, instance nodes are connected to their corresponding table nodes to bridge data and concepts.

3.2.2. Ontology Construction from Unstructured Data To harness the rich enterprise knowledge embedded in unstructured text, we construct the ontology $\mathcal{G}_U = (\mathcal{V}_U, \mathcal{E}_U)$ using a systematic pipeline comprising document parsing, triple extraction [?], entity disambiguation, and graph construction. Specifically, during the extraction phase, we deploy a locally hosted vLLM-accelerated Qwen3 [?] model (temperature set to 0.1) to process documents segmented into 2,000-token chunks with a 150-token overlap. This model is tasked with extracting entities along with their descriptive attributes and identifying relationships from the set $\{\text{MENTIONS}, \text{RELATES_TO}, \text{IS_A}\}$.

To resolve non-standard entity mentions and ambiguity, we apply layered merging rules: (1) initial surface matching via edit distance (≥ 0.85) and substring containment ($\geq 60\%$); (2) domain-specific normalization; and (3) deep semantic matching using BGE-M3 [?] embedding cosine similarity (≥ 0.85). This ensures synonymous mentions are consolidated into canonical nodes in \mathcal{V}_U .

The unified heterogeneous graph $\mathcal{G} = \mathcal{G}_S \cup \mathcal{G}_U$ is synthesized by integrating these two layers. We employ a heuristic tagging algorithm to generate attribute tags for file nodes in \mathcal{G}_U , which are subsequently matched against descriptions in the instance layer of \mathcal{G}_S to establish semantic associations $\Psi : \mathcal{V}_U \rightarrow \mathcal{V}_S$, effectively bridging the semantic gap between raw text and relational data. Furthermore, we utilize the link prediction capabilities of the LOM to infer missing relationships, thereby densifying and completing the graph structure.

4.1. Dataset

We develop a graph reasoning dataset for training and evaluation, which encompasses 19 diverse tasks across six categories: traversal, graph properties, node similarity, paths and flows, centrality, and tree structures. The data, stored in PyG format, incorporates 768-dimensional node features encoded by sentence-transformers and detailed edge structures. The training corpus consists of 95,000

samples supporting multi-stage instruction tuning, while the evaluation benchmark comprises 190 samples (10 per task type) stratified by difficulty: simple (e.g., BFS, node degree), medium (e.g., connectivity), and difficult (e.g., shortest path, PageRank). The graphs vary in size from 5 to 50 nodes and 5 to 200 edges, covering undirected, directed, and weighted types.

4.2. Hyper-parameters

We adopt Qwen3-4B-Instruct as the backbone LLM, featuring 4B parameters, a hidden size of 2,560, 36 layers, 32 attention heads, and a vocabulary of 151,936, with the maximum generation length set to 4,096 tokens. To capture graph structural information, we utilize a pre-trained graph transformer encoder consisting of 3 layers with a hidden dimension of 128 and 8 attention heads. Node semantics are initialized using the all-mpnet-base-v2 Sentence-Transformer to generate 768-dimensional features. These features are processed by the graph encoder via dimension-matching projection layers and finally mapped to the LLM's 2,560-dimensional embedding space through a linear graph projector.

4.3. Evaluation Metrics

We evaluate performance using accuracy as the primary metric on a benchmark comprising 190 test samples, evenly distributed across 19 distinct graph reasoning tasks. These tasks span six categories: traversal (BFS, DFS), graph properties (node degree, neighbor query, edge existence, connectivity, cycle detection, bipartiteness, connected components, diameter), node similarity (common neighbors, Jaccard similarity), paths and flows (predecessor, topological sort, shortest path, maximum flow), centrality (PageRank, clustering coefficient), and tree structure (minimum spanning tree). For assessment, we apply strict task-specific correctness criteria: exact order for traversals, precise values for numerical tasks, correct classifications for boolean queries, set matching for list outputs, and validity checks for path-related tasks.

4.4. Main Results

As shown in Figure 2 [Figure 2: see original paper], LOM-4B demonstrates superior performance across the benchmark, achieving the top rank with an overall accuracy of 89.47% (170/190 correct). This represents a substantial improvement over competing approaches, validating the effectiveness of our proposed method in handling complex graph reasoning tasks. The model's architecture, which integrates a graph encoder with a powerful large language model, evidently provides a distinct advantage in interpreting and solving graph-structured problems.

The comparison further highlights the performance gap between specialized and general-purpose models. The second-tier models, including DeepSeek-V3.2 [?], GraphInstruct [?], and Doubao-1.8, achieve comparable results in the range of 78.42% to 79.47%, yet they trail LOM-4B by approximately 10%. Notably, the

base Qwen3 models (Qwen-Max and Qwen3-4B-Instruct) exhibit significantly lower accuracies of 57.37% and 18.95% respectively, underscoring the necessity of targeted instruction tuning and structural encoding for effective graph reasoning execution.

Figure 2: Model Performance Ranking

4.5. Analysis

As shown in Figure 3 [Figure 3: see original paper], we present a detailed breakdown of model performance across 19 specific graph tasks. On fundamental graph traversal and property retrieval tasks—such as BFS, DFS, neighbor queries, and degree calculation—most advanced models including LOM-4B, DeepSeek-V3.2, and GraphInstruct achieve near-perfect accuracy (100%). This indicates that current LLMs, when equipped with appropriate training or structural awareness, have successfully mastered the basic syntax and local connectivity rules of graph data.

However, a significant divergence in performance emerges on computationally intensive and global reasoning tasks. LOM-4B exhibits remarkable robustness on algorithmic challenges that baffle other models. Specifically, for PageRank and minimum spanning tree (MST) tasks, where other leading models like DeepSeek-V3.2 and GraphInstruct score near 0%, LOM-4B maintains high accuracy (80% and 60% respectively). Similarly, in topological sorting, LOM-4B achieves 100% accuracy, outperforming the next best competitor by a margin, demonstrating its superior capacity to reason about global graph structures and dependencies.

The results also shed light on the impact of training stages and model specialization. Comparing LOM-Stage2, LOM-Stage3, and the final LOM-4B reveals a clear progression in capability, particularly for complex tasks like bipartite matching and shortest paths. While general-purpose models like Doubao-1.8 and DeepSeek-V3.2 show sporadic strengths (e.g., 100% in maximum flow or shortest path), they lack the consistent versatility of LOM-4B. The contrast with Qwen-Max further emphasizes that scale alone is insufficient for graph reasoning; the specialized architecture and instruction tuning of LOM are critical for bridging the gap between simple retrieval and complex algorithmic execution.

Figure 3: Model Performance Heatmap

4.6. Ablation Study

As shown in Figure 1, we conduct an ablation study to quantify the contribution of each key component in LOM-4B. The full model achieves the highest accuracy of 89.47%. Removing the CoT reasoning mechanism leads to a significant drop of over 11%, reducing accuracy to 78.42%. This indicates that explicit reasoning steps are crucial for guiding the model through complex multi-step

graph reasoning tasks, preventing it from jumping to incorrect conclusions and ensuring more robust problem-solving strategies.

The impact of instruction tuning and structural encoding is even more profound. Without instruction tuning (“w/o Instruct”), performance further degrades to 61.66%, highlighting the importance of aligning the LLM with specific graph-related tasks. Most strikingly, the baseline model without both the graph encoder and instruction tuning (“w/o GNN, Instruct”) collapses to a mere 18.95%. This drastic decline confirms that standard LLMs, lacking specific structural embeddings and task adaptation, are fundamentally ill-equipped for graph reasoning, validating the necessity of our proposed hybrid architecture.

Table 1 Ablation Study Results

Settings	Accuracy
LOM-4B	89.47%
w/o CoT	78.42%
w/o Instruct	61.66%
w/o GNN, Instruct	18.95%

5. Conclusion

We have presented the LOM, a unified framework for enterprise ontology construction and reasoning that effectively bridges the gap between structured databases and unstructured textual knowledge. By integrating a dual-layer ontology construction method with a three-stage instruction alignment pipeline—spanning ontology instruction fine-tuning, text-ontology grounding, and multi-task instruction tuning—we enable the model to perform complex, structure-aware reasoning over heterogeneous enterprise data. Our experiments demonstrate that the LOM-4B achieves state-of-the-art performance on diverse ontology reasoning tasks, proving its efficacy in handling the intricacies of real-world enterprise environments.

In future work, we plan to enhance the performance of LOM on complex ontology reasoning tasks, particularly focusing on challenging tasks like MST and PageRank where current models consistently achieve low scores. We plan to adopt more reinforcement learning strategies to enable deeper structural understanding and more accurate multi-hop inference to overcome these limitations.

References

- [1] Assche, D.V., Jankaj, J., Meester, B.D., 2024. Backwards or forwards? [R2]RML backwards compatibility in rmlmapper, in: Chaves-Fraga, D., Dimou, A., Iglesias-Molina, A., Serles, U., Assche, D.V. (Eds.), Proceedings of the 5th International Workshop on Knowledge Graph Construction co-located with 21th

Extended Semantic Web Conference (ESWC 2024), Hersonissos, Greece, May 27, 2024, CEUR-WS.org.

[2] Bai, J., Fan, W., Hu, Q., Zong, Q., Li, C., Tsang, H.T., Luo, H., Yim, Y., Huang, H., Zhou, X., Qin, F., Zheng, T., Peng, X., Yao, X., Yang, H., Wu, L., Ji, Y., Zhang, G., Chen, R., Song, Y., 2025. Autoschemakg: Autonomous knowledge graph construction through dynamic schema induction from web-scale corpora. CoRR abs/2505.23628. doi:10.48550/ARXIV.2505.23628, arXiv:2505.23628.

[3] Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., Liu, Z., 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv preprint arXiv:2402.03216 4.

[4] Dai, X., Hua, Y., Wu, T., Sheng, Y., Ji, Q., Qi, G., 2025. Large language models can better understand knowledge graphs than we thought. Knowl. Based Syst. 312, 113060. doi:10.1016/J.KNOSYS.

[5] Hou, Z., He, Y., Cen, Y., Liu, X., Dong, Y., Kharlamov, E., Tang, J., 2023. Graphmae2: A decoding-enhanced masked self-supervised graph learner, in: Proceedings of the ACM web conference 2023, pp.

[6] Iglesias, E., Jozashoori, S., Chaves-Fraga, D., Collarana, D., Vidal, M., 2020. Sdm-rdfizer: An RML interpreter for the efficient creation of RDF knowledge graphs, in: d' Aquin, M., Dietze, S., Hauff, C., Curry, E., Cudré-Mauroux, P. (Eds.), CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, ACM. pp. 3039-3046. doi:10.1145/3340531.3412881.

[7] Li, Y., Zhang, C., Wu, C., Chen, J., Liu, Y., Zhu, H., 2026. Pre-training graph autoencoder incorporating hierarchical topology knowledge. Data Intelligence doi:10.3724/2096-7004.di.2026.0001.

[8] Liu, A., Mei, A., Lin, B., Xue, B., Wang, B., Xu, B., Wu, B., Zhang, B., Lin, C., Dong, C., et al., 2025. Deepseek-v3.2: Pushing the frontier of open large language models. arXiv preprint arXiv:2512.02556 .

[9] Luo, L., Zhao, Z., Liu, J., Qiu, Z., Dong, J., Panev, S., Gong, C., Vu, T., Haffari, G., Phung, D.Q., Liew, A.W., Pan, S., 2025. G-reasoner: Foundation models for unified reasoning over graph-structured knowledge. CoRR abs/2509.24276. doi:10.48550/ARXIV.2509.24276, arXiv:2509.24276.

[10] Luo, Z., Song, X., Huang, H., Lian, J., Zhang, C., Jiang, J., Xie, X., 2024. Graphinstruct: Empowering large language models with graph understanding and reasoning capability. CoRR abs/2403.04483. doi:10.48550/ARXIV.2403.04483, arXiv:2403.04483.

[11] Mavromatis, C., Karypis, G., 2025. GNN-RAG: graph neural retrieval for efficient large language model reasoning on knowledge graphs, in: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (Eds.), Findings of the Association

for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, Association for Computational Linguistics. pp. 16682-16699.

[12] Mendes, R., Oliveira, D., Garcia, V., 2024. Application of generative ai as an enterprise wikibase knowledge graph q&a system, in: Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024), pp. 35-42.

[13] Nayyeri, M., Yogi, A.A., Fathallah, N., Thapa, R.B., Tautenhahn, H., Schnurpel, A., Staab, S., 2025. Retrieval-augmented generation of ontologies from relational databases. CoRR abs/2506.01232. doi:10.48550/ARXIV.2506.01232, arXiv:2506.01232.

[14] Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X., 2024. Unifying large language models and knowledge graphs: A roadmap. Trans. Knowl. Data Eng. 36, 3580-3599. doi:10.1109/TKDE.2024.

[15] Tan, X., Wang, X., Liu, Q., Xu, X., Yuan, X., Zhang, W., 2025. Paths-over-graph: Knowledge graph empowered large language model reasoning, in: Long, G., Blumstein, M., Chang, Y., Lewin-Eytan, L., Huang, Z.H., Yom-Tov, E. (Eds.), Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025, ACM. pp. 3505-3522. doi:10.1145/3696410.

[16] Tang, J., Yang, Y., Wei, W., Shi, L., Xia, L., Yin, D., Huang, C., 2024. Hight: Heterogeneous graph language model, in: Baeza-Yates, R., Bonchi, F. (Eds.), Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024, ACM. pp. 2842-2853. doi:10.1145/.

[17] Trajanoska, M., Stojanov, R., Trajanov, D., 2025. A multi-agent system for semantic mapping of relational data to knowledge graphs. arXiv preprint arXiv:2511.06455 .

[18] Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al., 2025. Qwen3 technical report. arXiv preprint arXiv:2505.09388 .

[19] Yang, Y., Tang, J., Xia, L., Zou, X., Liang, Y., Huang, C., Graphagent: Agentic graph language assistant. abs/2412.17029. URL: <https://doi.org/10.48550/arXiv.2412.17029>, doi:10.48550/ARXIV.2412.17029, arXiv:2412.17029.

[20] Zhou, Y., Li, W., Lu, Y., Li, J., Liu, F., Zhang, M., Wang, Y., He, D., Liu, H., Zhang, M., 2025. Reflection on knowledge graph for large language models reasoning, in: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (Eds.), Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, Association for Computational Linguistics. pp. 23840-23857.

[21] Zhu, H., 2024. Node classification via semantic-structural attention-enhanced graph convolutional networks. CoRR abs/2403.16033. doi:10.48550/ARXIV.2403.16033,

arXiv:2403.16033.

[22] Zhu, H., Peng, H., Lyu, Z., Hou, L., Li, J., Xiao, J., 2023. Pre-training language model incorporating domain-specific heterogeneous knowledge into a unified representation. *Expert Systems with Applications* 215, 119369.

[23] Zhu, H., Tiwari, P., Zhang, Y., Gupta, D., Alharbi, M., Nguyen, T.G., Dehdashti, S., 2022. Switchnet: A modular neural network for adaptive relation extraction. *Comput. Electr. Eng.* 104, 108445. doi:10.1016/J.COMPELECENG.2022.108445.

[24] Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., Deng, S., Chen, H., Zhang, N., 2024. Llms for knowledge graph construction and reasoning: recent capabilities and future opportunities. *World Wide Web (WWW)* 27, 58. doi:10.1007/S11280-024-01297-W.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv –Machine translation. Verify with original.