Journal Title
XX(X):1–30
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

# Retrieval-Augmented Ontologic Graph Model with GNN Integration to Enhance Inductive Reasoning

**Anonymous Authors(s)**[1]

**Abstract**

This paper introduces a novel e-learning knowledge retrieval method, significantly enhancing information access through the integration of ontology and graph-based reasoning. Ontology provides a structured framework for domain knowledge, while knowledge graphs visualize the intricate connections between concepts. By employing graph-based reasoning, the designed system can infer relations between interrelated topics to deliver deeper, more contextually relevant answers to user queries. This approach effectively addresses the challenge of knowledge gaps within e-learning materials. The model demonstrates the ability to deduce new information from existing data, bridging these gaps and enriching the learning experience. Experimental results confirm the superior performance of the proposed method compared to existing techniques, showcasing its capacity to provide more accurate and comprehensive information. To validate its practical application, it has been applied to develop an intelligent assistance system for the *Database Systems* course. This real-world implementation demonstrates the system's potential to revolutionize e-learning, offering personalized support and fostering a more effective and engaging learning environment. The integration of ontology and knowledge graphs, coupled with graph-based reasoning, presents a significant advancement in e-learning knowledge retrieval.

**Keywords**

Knowledge Engineering; Ontology; Graph Neural Network; Symbolic AI; Automated reasoning; Neurosymbolic

[1]Anonymous Institution(s)

**Corresponding author:**
Anonymous Author(s)
Email: anonymous@sage.co.uk

*Prepared using* sagej.cls *[Version: 2017/01/17 v1.20]*

## Introduction

Ontology also provides a structured way to represent domain knowledge and they are also useful to handling multi-step reasoning due to their reliance on static rules[1]. Knowledge graphs extend ontologies by representing both concepts and their relationships within a flexible, graph-based structure[2,3]. This allows for exploring relationships not explicitly defined by an ontology; however, knowledge graphs still struggle to capture deeply interdependent and hierarchical relationships. This is particularly challenging in subjects like *Fundamentals of Database Systems*, where key concepts such as transactions, normalization, and data integrity are highly interconnected and context-dependent. Consider the following scenario as:

**Example 1**: "What happens to a transaction when a deadlock is detected?"

Answering this question requires an understanding of not just the definitions of transactions and deadlocks, but also how these concepts interact within the broader context of database systems. A static rule-based system would struggle to infer the necessary relationships to provide a comprehensive answer. In contrast, a graph-based reasoning system can dynamically trace the connections between *"Transactions," "Deadlocks"*, and *"Rollbacks"* to deliver an insightful response. This capability is illustrated in Figure 1.

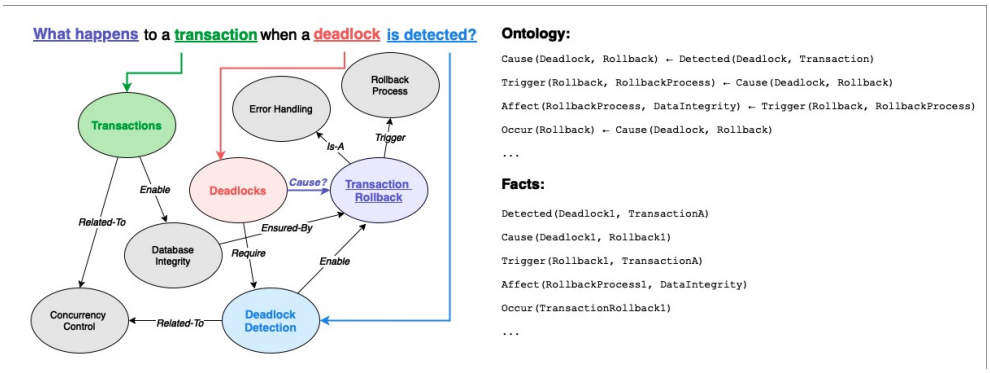



**Figure 1.** Integration of knowledge graph and ontology model reasoning for Example 1.

In this study, a GNN-based Retrieval-Augmented Ontologic Graph Model is proposed combining the structured knowledge representation of ontology Rela-Ops model[1,4] with the flexibility of knowledge graphs. The proposed model is called *Rela-GNN* model. By integrating graph-based reasoning with a dynamic retrieval mechanism, the proposed model enhances the inductive reasoning capabilities[5]. This approach enables the system to bridge knowledge gaps, infer deeper connections between concepts, and provide more contextually accurate responses to student queries[6]. A case study on the *Database Systems* course demonstrates the effectiveness of our approach in improving reasoning capabilities, enhancing knowledge retrieval, and supporting deeper learning outcomes.

This paper is an extended version of the study in[7]. In this study, Rela-GNN combines explicit knowledge representation with neural relational learning. Rela-Onto provides hierarchical concepts and typed relationships, while the GNN learns relational patterns to support knowledge retrieval. The connection between these components is established through Rela-WNet, where semantic, frequency,

and hierarchical information guide relation weighting and path selection. This ontology-guided neural reasoning approach also provides a basis for interpreting retrieval results through the concepts and relationships contained in the retrieved subgraphs.

This paper significantly expands the study in [7] by introducing three novel contributions. First, the Rela-GNN framework is evaluated on the multi-relational NELL-995 dataset, demonstrating its inductive reasoning robustness. Second, a quantitative link prediction analysis is introduced for specifically applying to Database Foundation educational knowledge graph in both transductive and inductive settings. Finally, we thoroughly explore graph-based retrieval interpretability and isolate the LLM's role, confirming that educational value stems primarily from accurate knowledge retrieval rather than just language fluency.

## Related Works

The integration of ontologies and knowledge graphs for knowledge representation and retrieval has been widely researched across domains such as education, artificial intelligence, and information systems [8–10]. These methods are effective for building intelligent systems in education [11,12]. Despite their potential, existing approaches face limitations in effectively supporting learning in complex subjects like *Fundamentals of Database Systems*. These limitations stem from their static structures and lack of dynamic reasoning capabilities.

Ontologies are traditionally used to structure educational content and define relationships between concepts, ensuring semantic consistency [13–15]. However, ontology-based frameworks often rely on rigid structures and predefined rules, which limits their ability to model dynamic and context-dependent relationships. For instance, in database systems, where topics like normalization, functional dependencies, and transaction management are intricately linked, static ontologies struggle to adapt to diverse and evolving learning needs [16].

Knowledge graphs have advanced the representation of interconnected concepts to improve search and retrieval [17,18]. In educational settings, knowledge graphs have been used to support adaptive learning environments and improve navigation between related topics [19]. However, these systems often focus on organizing and presenting information rather than enabling multi-step reasoning. This is a critical limitation in subjects like database systems, where understanding requires exploring the interplay between concepts.

Recent efforts have combined ontologies and knowledge graphs to improve information retrieval within intelligent systems [20,21]. However, these approaches still emphasize the static organization of knowledge rather than supporting dynamic, inferential reasoning. Although a knowledge graph can illustrate the relationships between database concepts, without dynamic reasoning, it cannot fully answer complex queries that require multi-step inference.

Graph-based reasoning has emerged as a promising solution to this problem by leveraging the relationships in knowledge graphs to infer new information [22,23]. This approach enables systems to discover hidden connections that are not explicitly defined. Although studies have begun to explore graph-based reasoning in education [24], existing models remain inadequate for the course as *Database Systems*, where answering questions often requires reasoning across multiple interconnected concepts.

The research in [25,26] has applied knowledge graphs to personalized learning in MOOCs, primarily for content recommendation. Similarly, the study in [27] explored graph-based approaches in English majors

to improve problem-solving. However, these efforts focus more on content navigation and suggestion than on enabling deep, interconnected reasoning. In subjects like database systems, this gap becomes evident.

The shortcomings of existing ontology and knowledge graph models highlight the need for a more dynamic and flexible approach to reasoning in educational systems. Relying solely on static rule-based or pre-defined graph structures limits the ability to provide meaningful insights in complex, interdependent domains. This is particularly relevant for *Fundamentals of Database Systems*, where understanding interconnected topics is critical for mastering the subject.

To address these limitations, we propose *Rela-GNN* model which combines graph-based reasoning and dynamic knowledge retrieval. This model enhances inductive reasoning by enabling the system to infer deeper connections between related concepts, thus providing a more intuitive and context-aware learning experience. By moving beyond static knowledge organization, this approach allows students to explore and understand interconnected concepts in a deeper, more meaningful way.

## Ontology and Knowledge Graph Integration

This section introduces two modeling contributions: Rela-Onto for hierarchical concept organization and Rela-WNet for weighted relational representation. Together, they provide the knowledge structure that Rela-GNN reasons over.

### *Hierarchical Ontology Model based on Rela-Onto*

Prior work on the Onto-Data model[28] combined Rela-Ops with structured databases for educational querying. That model organizes domain knowledge through semantic representation and efficient storage[29]. However, it lacks explicit support for hierarchical relationships between concepts—a gap that limits its ability to represent how course content is typically structured (chapters containing lessons, lessons containing definitions, etc.).

We extend Onto-Data to address this limitation. The extension, called *Rela-Onto*, adds hierarchical concept layers and typed relationships:

**Definition 3.1.1**[7]**:** *Onto-Data* is an integrating model of the extension ontology **Rela-Ops** and the structure of a database, includes:

$$\mathcal{K} = (\mathbf{C}, \mathbf{R}, \mathbf{Ops}, \mathbf{Rules}, \mathbf{Problems}, \mathbf{Methods}) \oplus \mathbf{D}_{\text{data}}$$

In which, (**C**, **R**, **Ops**, **Rules**, **Problems**, **Methods**) follows the Rela-Ops structure: **C** contains concepts (object classes), **R** defines relationships (*is-a*, *has-a*, *related-to*), **Ops** provides operators, **Rules** captures domain inference rules, **Problems** stores exercises, and **Methods** holds solution strategies. The database $\mathbf{D}_{\text{data}}$ stores actual content. See[20] for details.

Onto-Data has the limitation. It does not distinguish concept abstraction levels or relationship types. We introduce Rela-Onto to address this:

**Definition 3.1.2**[7]**:** *Rela-Onto* - the extension ontology of **Onto-Data** for hierarchical knowledge organization is defined as follows:

$$\mathcal{K}_{hier} = (\mathbf{C_H}, \mathbf{R_H}, Ops, Rules, Problems, Methods) \oplus \mathbf{D}_{\text{data}}$$

$\mathbf{C_H}$: A set of hierarchical concepts organized into multiple abstraction layers. This hierarchical structure aims to capture and organize domain knowledge effectively.

$$\mathbf{C_H} = \{\mathrm{C}_{h \mid h \in \{hi,mi,lo\}} \mid \mathrm{C}_h = (\mathit{Attrs}, \mathit{Properties})\}$$

where, *hi*, *mi*, *lo* stands for *high*, *mid*, *low* level concepts. Each $C_h \in \mathbf{C_H}$ is a tube (*Attrs*, *Properties*), in which, *Attrs* is a set of additional information of the concept such as descriptions, examples, and any domain-specific metadata. *Properties* is a set of core identifying characteristics of the concept.

$\mathbf{R_H}$: A set of hierarchical relationships organized into multiple interaction types. This relational structure aims to effectively capture and organize the connections between hierarchical concepts.

$$\mathbf{R_H} = \{\mathrm{R}_{h \mid h \in \{ve,ho,cr\}} \mid \mathrm{R}_h = (\mathit{Start}, \mathit{End}, \mathit{Properties}, \mathit{Attrs})\}$$

where, *ve*, *ho*, *cr* stands for *vertical*, *horizontal*, *cross-hierarchical* relationships between concepts. Each $R_h \in \mathbf{R_H}$ is a tube (*Start*, *End*, *Attrs*, *Properties*), in which, *Start* and *End* represent the starting and ending concepts - $C_s$ and $C_e$ of the relationship, denotes as $C_s \xrightarrow{R_h} C_e$. *Attrs* is a set of additional information of the relationship such as descriptions, examples, and any domain-specific metadata. *Properties* is a set of core identifying characteristics of the relationship. Table 1 describes hierarchical conceptual of Concepts/Relationships and illustrates some examples in the knowledge of *Fundamental of Database Systems* course.

**Table 1.** Describe each level of the hierarchical organization of concepts and relations.

| | Hierarchy | Description | Examples |
|---|---|---|---|
| **Concept Level** | **High-level** | Broad, overarching concepts covering general knowledge categories. | *Normalization*, *Data Integrity*, *Algorithm Complexity* |
| | **Mid-level** | Sub-divisions of high-level concepts focusing on more specific areas. | *First Normal Form (1NF)*, *NP-Completeness* |
| | **Low-level** | Specific details or principles related to mid-level concepts. | *1NF Properties*, *Functional Dependencies* |
| **Relationships** | **Vertical** | Connections between concepts at different abstraction levels. | *Normalization* $\rightarrow$ *First Normal Form (1NF)* $\rightarrow$ *1NF properties* |
| | **Horizontal** | Connections between concepts at the same level but within different branches. | *Second Normal Form (2NF)* $\leftrightarrow$ *Boyce-Codd Normal Form (BCNF)* |
| | **Cross-Hierarchical** | Relationships that cross between layers or branches, such as dependencies between higher and lower-level concepts. | *Algorithm Complexity* $\leftrightarrow$ *Optimized Data Structures* |

In the context of database systems, under the framework for organizing database knowledge, the structure of database topics can be categorized into three levels: foundational concepts, core principles, and advanced methodologies. The deepest and most complex structure of database systems knowledge is organized into five levels, which include module, chapter, lesson, concept, and point. According to the course framework for Database Systems, the sample knowledge of these levels are as Figure 2:

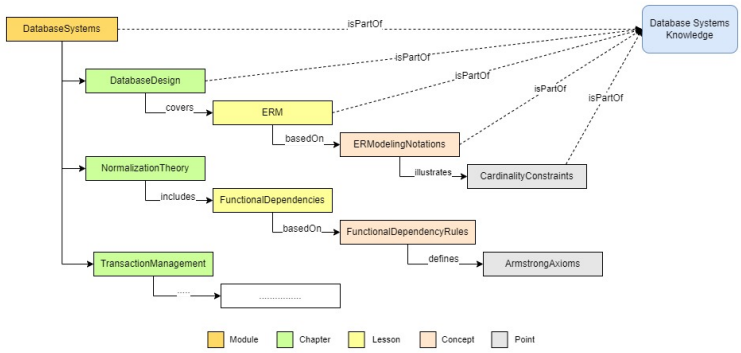



**Figure 2.** Knowledge of *Fundamentals of Database Systems* course represented by Rela-Onto.

## *Rule-based reasoning Ontologic Graph*

Integrating a knowledge graph into an ontology is essential for enhancing the representation and organization of domain-specific knowledge[2,30]. While an ontology provides a structured and hierarchical framework, a knowledge graph enriches it by incorporating dynamic and contextual connections[23]. This integration facilitates inductive reasoning within educational chatbots, enabling them to infer deeper relationships and respond more effectively to complex queries.

A study by[20] demonstrates the effectiveness of using a two-layer knowledge graph in educational systems through the **Rela-KG** model. A key strength of this model is its use of simple rule-based reasoning, leveraging relationships and rules within the ontology. This approach allows the system to perform logical inferences and answer complex queries without the need for advanced algorithms. The structure of this model is described in the following definition:

**Definition 3.2.1:** *Rela-KG* is an integrated model combining the extension ontology **Rela-Ops** with a two-layer knowledge graph structure. It is formally defined as:

$$\mathcal{K} = (\mathbf{C}, \mathbf{R}, \mathbf{Ops}, \mathbf{Rules}) \oplus (\mathbf{Problems}, \mathbf{Methods}) \xrightarrow{extract} \mathcal{K}_{KG} = (\mathbf{V}, \mathbf{E})$$

where, $\mathbf{V}$ is a set of *vertices* expressing the knowledge, $\mathbf{E}$ is a set of *edges* representing relations between $v \in \mathbf{V}$. The model distinguishes between two types of edges: $E_{\text{attrs}}$ and $E_{\text{objs}}$, enabling a clear separation between attribute-based relations within an object and relations connecting different objects.

The model is organized into three connected aspects, each contributing to different dimensions of knowledge representation:

- **Knowledge:** Contains nodes representing concepts, problems, and operators, providing the core domain knowledge.
- **Relations:** Represents the relationships between concepts, linking nodes through edges that illustrate their connections within the domain.
- **Rules:** Contains nodes that hold logical rules and methods, which help in reasoning and deriving conclusions from the knowledge base.

However, relying solely on simple rule-based reasoning has its limitations[31]. To demonstrate this, let's consider the following example from the *Fundamentals of Database Systems* course.

**Example 2:** Given a set of concepts $\mathbf{C}_{ex}$ from the *Fundamentals of Database Systems* course including: Primary Key (*PKey*), First Normal Form (*1NF*), Functional Dependencies (*FuncDep*), Boyce-Codd Normal Form (*BCNF*)

$$\mathbf{C}_{ex} := \{PKey, 1NF, FuncDep, BCNF\}$$

To represent the traditional rule-based system using the concept of a **Rules**-set for prerequisite and suggestion knowledge, we can frame it as:

| **Rules:** | **Description:** |
|---|---|
| `suggest(1NF) ← understand(PKey)` | If a student understands *Primary Key*, then suggest *First Normal Form*. |
| `suggest(FuncDep) ← understand(1NF)` | If a student understands *First Normal Form*, then suggest *Functional Dependencies*. |
| `suggest(BCNF) ← understand(FuncDep)` | If a student understands *Function Dependencies*, then suggest *Boyce-Codd Normal Form*. |

**Figure 3.** Illustrates the rule-based reasoning of **Example 2**.

With these rules, the model can easily reasoning the simple direct rules without logical inference. However, suppose the student struggles with a problem related to *BCNF* involving transitive dependencies. The rule-based system might only suggest reviewing *Functional Dependencies*. The static rules might not account for the nuanced interdependencies between *Primary Key*, *1NF*, and other aspects of normalization, such as multivalued dependencies. The rule-based reasoning fails in this case because:

- **Linear relationships:** It assumes a linear progression through the concepts (*Primary Key* $\rightarrow$ *1NF* $\rightarrow$ *Functional Dependencies* $\rightarrow$ *BCNF*), without understanding the deeper connections between these concepts.
- **Lack of context-awareness:** The system doesn't capture complex relationships or context, like the importance of understanding *Primary Key* thoroughly before mastering *1NF* and progressing to more advanced normal forms.
- **Inability to identify knowledge gaps** The rule-based approach doesn't detect that missing the understanding of *Primary Key* could impact grasping *BCNF*.

To address these challenges, we proposed a GNN-based reasoning framework designed to enhance the model's inductive learning capabilities. This framework enables the model to capture more intricate relationships and infer indirect dependencies effectively. However, to align this new framework with our knowledge representation model, modifications to the Rela-KG model are necessary to represent a richer set of relationships between nodes.

## *Rela-WNet: An extension of Rela-KG model*

In order to capture more intricate relationships between nodes, we designed *Rela-WNet* - an extension of Rela-KG model. We utilize *Rela-Onto* model as the knowledge representation ontology, while evaluate more the structure of the knowledge graph:

$$\mathcal{K}_{KG} = \textit{Rela-Onto} \oplus (\mathbf{H}, \mathbf{R}, \mathbf{T})$$

In which, **H** and **T** are the sets of head and tail entities, representing the extracted domain knowledge component, and **R** is the set of relations between entities. Additionally, to capture diverse and dynamic relationships between entities, **R** is divided into two primary sets:

$$\mathbf{R} = \mathbf{R}_{\text{intra}} \cup \mathbf{R}_{\text{infer}}$$

- $\mathbf{R}_{intra}$ is a set of edges that can model more sophisticated intra-object relations such as *hierarchical dependencies*, *cyclical connections*, and *conditional bindings* **within an object**.
- $\mathbf{R}_{infer}$ is a set of edges that capture multi-layered relationships such as *indirect influences*, *collaborative interactions*, and *transitive relations* **between different objects**.

Each relation $r \in \mathbf{R}$ is defined to capture semantic connection between entities: (*Hier*, *Weight*, *Context*, *Start*, *End*). In which, *Hier* specifies the hierarchical type of relationship $h_r \in (ve, ho, cr)$. *Weight* quantifies the significance of connections between nodes, we will describe in Section . *Context* is the semantic information that affects the relationship. *Start* and *End* are the head $h \in \mathbf{H}$ and tail $t \in \mathbf{T}$ of relation $r$, denotes as $h \xrightarrow{r} t$.

## GNN-Based Framework for Inductive Reasoning

### *Approach Overview*

GNNs have proven effective for relation prediction on knowledge graphs[31,32]. Our approach builds on Relational Graph Convolutional Networks (R-GCN)[33], using LogCo[34] as the backbone due to its strong performance on inductive tasks[35,36].

LogCo introduces entity-independent semantics through contrastive representations, but it treats all relational paths equally. This limitation matters for educational applications where some concept relationships are more important than others. LogCo is extended by integrating it with Rela-WNet (Section ), which assigns learned weights to relations based on their semantic and structural properties.

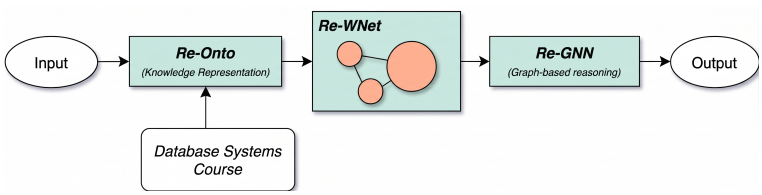



**Figure 4.** Overview of inductive reasoning approach.

Figure 4 illustrates the overall pipeline. Given a user query, we first extract entities and relations using standard NLP techniques, then map these to nodes in the knowledge graph. Rela-GNN performs message passing over the graph, aggregating information from neighboring nodes while weighting edges by their learned importance. The resulting node embeddings guide subgraph retrieval, and the retrieved content is passed to an LLM for response generation.

The key departure from prior work is the weighted aggregation step. Rather than treating all incoming edges equally, Rela-GNN prioritizes paths containing high-weight relations, those identified as semantically and pedagogically significant. Section  details this retrieval mechanism.

## *Relation Weighting*

We now describe how Rela-GNN computes edge weights and uses them during path selection.

*Weight Computation* For a triple $(h, r, t)$ in knowledge graph $G = \{R, E, T\}$, we compute the weight $w_r$ of relation $r$ from three signals: the semantic embedding score between $h$ and $t$, the frequency of $r$ in $G$, and the hierarchical type of $r$.

**Relational embeddings.** We utilize a ComplEx-based[37] for obtaining embeddings of entities in $G$. Formally, for a head entity $h$, tail entity $t$, and relation $r$, the embeddings are defined as: $f(h) = \mathbf{h}^{\text{Re}} + i \cdot \mathbf{h}^{\text{Im}}$, $f(t) = \mathbf{t}^{\text{Re}} + i \cdot \mathbf{t}^{\text{Im}}$, and $f(r) = \mathbf{r}^{\text{Re}} + i \cdot \mathbf{r}^{\text{Im}}$. The scoring function measures the similarity between entities and relations by taking the real part of the multi-linear product, given by:

$$\sigma'(h, r, t) = \Re\left(\langle f(h), f(r), \overline{f(t)}\rangle\right) \tag{1}$$

where $\overline{f(t)}$ denotes the complex conjugate of $f(t)$. This formulation allows us to capture richer relationships in $G$.

**Relational Frequency.** The frequency of relation $r$ is calculated based on its occurrence across all triples in $G$. Formally, for a relation $r$, its frequency weight is defined as: $\varphi(r) = \frac{\sum_{(h,r,t)\in T} \mathbf{1}_r}{\sum_{(h',r',t')\in T} \mathbf{1}}$, where $\mathbf{1}_r$ is an indicator function that counts the occurrence of the relation $r$ in triple $e$, and the denominator represents the total number of relations in the set of all triples $T$. To normalize the frequency weight, we compute the normalized frequency weight as:

$$\nu(r) = \frac{\varphi(r)}{\max_{r' \in R} \varphi(r')} \tag{2}$$

**Relation Weight.** To determine the weight of a relation $r$ in the KG $G$, we combine its semantic embedding, frequency score, and hierarchical attribute. Formally, the weight $w(r)$ of a relation $r$ is defined as:

$$w(r) = \lambda_1 \cdot \sigma'(h, r, t) + \lambda_2 \cdot \nu(r) + \lambda_3 \cdot \text{hier}(r), \tag{3}$$

where $\sigma'(h, r, t)$ represents the similarity score derived from the relational embedding from Eq. (1), and $\nu(r)$ denotes the normalized frequency score from Eq. (2). The hierarchical attribute $\text{hier}(r)$ is adjusted based on the relation type, with coefficients $\alpha_{ve}$, $\alpha_{ho}$, and $\alpha_{cr}$. The values of $\lambda_1$, $\lambda_2$, and $\lambda_3$ were chosen through hyperparameter tuning. To handle cases of zero or near-zero weights, we apply an inverse transformation with a small positive constant $\epsilon$ and cap the resulting values at a threshold $C$. The final normalized weight is given by:

$$w^*(r) = \frac{\min\left(\frac{1}{w(r)+\epsilon}, C\right) - \min(W)}{\max(W) - \min(W)} \tag{4}$$

where $W = \left\{ \min\left(\frac{1}{w(r)+\epsilon}, C\right) \mid r \in R \right\}$ represents the set of capped transformed weights for all relations $r \in R$.

After computing relation weights $w^*(r)$, the next step is to incorporate these weights into the process of extracting and selecting paths.

*Weighted Contrastive Path Generation* Building upon the computed relation weights, we aim to integrate these weights into the path extraction process. The goal is to enhance path selection by prioritizing those that not only maintain topological consistency but also align with the semantic significance of relations. This refined strategy enables the model to leverage weighted relations for more effective contrastive learning.

**Weighted Path Extraction.** Given a target triple $e$ as $(h, r, t)$, we construct an enclosing subgraph $G_T$ centered around the head entity $h$ and tail entity $t$. The extraction of weighted relational paths involves finding all paths from $h$ to $t$ in $G_T$ whose cumulative weight is no greater than a predefined threshold $L_{max}$. We achieve this by utilizing Dijkstra's algorithm, which efficiently identifies the shortest paths based on the sum of edge weights between $h$ and $t$, as shown in Algorithm 1. The set of extracted weighted paths is denoted as $\mathcal{P}_{h\to t}$.

**Input:** $G_T = (V, E), h, t, w(e)\ \forall e \in E$
**Output:** Set of paths $\mathcal{P}_{h\to t}$
**Initialization:**
  $Q \leftarrow \{(h, 0)\}$ ; `// Priority queue initialized with source node` $h$ `and distance 0`
  $d[v] \leftarrow \infty\ \forall v \in V, \quad d[h] \leftarrow 0$ ; `// Set initial distances`
  prev$[v] \leftarrow$ null $\forall v \in V$ ; `// Initialize predecessors to track paths`
**while** $Q \neq \emptyset$ **do**
  $u \leftarrow$ extract_min$(Q)$ ; `// Extract node` $u$ `with smallest distance`
  **for** $v \in$ *neighbors*$(u)$ **do**
    **if** $d[u] + w(u, v) < d[v]$ **then**
      $d[v] \leftarrow d[u] + w(u, v)$ ; `// Update distance to node` $v$
      prev$[v] \leftarrow u$ ; `// Update predecessor of` $v$
      $Q \leftarrow Q \cup \{(v, d[v])\}$ ; `// Add` $v$ `to priority queue`
    **end**
  **end**
**end**
$\mathcal{P}_{h\to t} \leftarrow$ trace_back$(t,$ prev$)$ ; `// Construct path set by tracing back from` $t$
**return** $\mathcal{P}_{h\to t}$;

**Algorithm 1:** Dijkstra's Algorithm for Weighted Path Extraction

With the set of weighted paths extracted, the next step is to categorize these paths into positive and negative samples based on their alignment with the target relation's weight.

**Positive and Negative Path Sampling.** In the previous approach, positive and negative paths were selected solely based on topological consistency with the target triple $(h, r, t)$. The set of positive paths $\mathcal{P}^+_{h\to t}$ was defined as the collection of paths between $h$ and $t$ in the enclosing subgraph $G_T$, without considering the weights of relations. This led to a limitation where paths that were topologically

correct but semantically inconsistent (due to differing weights) could be incorrectly selected as positive samples. To address this issue, we redefine the selection criteria for both positive and negative paths by incorporating relation weights:

- **Positive Paths:** The set of positive paths $\mathcal{P}^+_{h\to t}$ consists of paths from $h$ to $t$ in $G_T$ such that the cumulative weight of the path $w(p)$ is close to the weight of the target relation $w(r)$. Specifically, a path $p$ from $h$ to $t$ is included in $\mathcal{P}^+_{h\to t}$ if it satisfies:

$$|w(p) - w(r)| \leq \delta \tag{5}$$

  where $w(p)$ denotes the cumulative weight of the path $p$, $w(r)$ is the weight of the target relation $r$ (computed as described in Section ), and $\delta$ is a predefined tolerance threshold. This ensures that the selected positive paths align both topologically and semantically with the target relation.
- **Negative Paths:** The set of negative paths $\mathcal{P}^-_{h\to t}$ is composed of paths from $h$ to $t$ that differ significantly in their cumulative weight from the target relation. A path $p$ is included in $\mathcal{P}^-_{h\to t}$ if it satisfies:

$$|w(p) - w(r)| > \delta \tag{6}$$

  Additionally, we introduce a random sampling strategy to select paths that are structurally different from positive paths, ensuring that $\mathcal{P}^-_{h\to t}$ contains paths that do not inadvertently share similar structural features with $\mathcal{P}^+_{h\to t}$. This prevents false positives and strengthens the contrastive learning process.

**Contrastive Learning Objective.** With the incorporation of relation weights into the selection of positive and negative paths, we aim to prioritize paths that are semantically consistent with the target relation. The updated contrastive loss function is defined as:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp\left(\mathcal{S}(r_T, p_k^+)\right)}{\exp\left(\mathcal{S}(r_T, p_k^+)\right) + \exp\left(\mathcal{S}(r_T, p_k^-)\right)} \tag{7}$$

where the similarity score $\mathcal{S}(r_T, p)$ is given by:

$$\mathcal{S}(r_T, p) = \text{ReLU}\left(\sigma'(h, r, t) - |w(p) - w(r)|\right) \tag{8}$$

This updated objective encourages the model to maximize the similarity between the target relation $r_T$ and positive paths $p_k^+$, while minimizing the similarity with negative paths $p_k^-$. By accounting for relation weights, the model effectively captures both topological and semantic consistency, resulting in improved path selection and relation prediction.

## *Retrieval Mechanism*

The proposed retrieval mechanism leverages the Rela-GNN framework to effectively extract and infer relevant information from the structured KG. This mechanism aims to address complex queries by combining graph traversal, contextual weighting of relations, and inductive reasoning. By modeling the relational dependencies and prioritizing contextually significant connections, the Rela-GNN framework

enhances the ability to retrieve accurate and contextually appropriate answers. Algorithm 2 outlines the detailed steps involved in the retrieval process:

**Input:** Query $Q$, Knowledge Graph $G = (V, E)$
**Output:** Relevant Subgraph $S$
**Step 1: Structured Query Formulation**
$E_Q, R_Q \leftarrow \text{NLP}(Q)$ `// Extract key entities and relations`
$N_{init} = \{v \in V \mid v \in E_Q\}$ `// Map entities to graph nodes`
**Step 2: Initial node Identification**
$N_{init} \subseteq V$ `// Identify relevant nodes`
**Step 3: Aggregate information from neighboring nodes**
**foreach** $v_i \in N_{init}$ **do**
  $\mathbf{h}_i^{(l+1)} \leftarrow \sigma\left(\sum_{r\in R}\sum_{v_j\in\mathcal{N}_r(v_i)} w_{r,ij} \cdot \mathbf{W}_r^{(l)} \cdot \mathbf{h}_j^{(l)}\right)$
**end**
**Step 4: Rank paths based on weights**
$\mathcal{P}_{ranked} = \text{Rank}\left(\{(v_i, r, v_j) \mid v_i, v_j \in N_{init}, r \in R\}\right)$
**Step 5: Subgraph Selection**
$S \leftarrow$ Select top-$k$ paths from $\mathcal{P}_{ranked}$
**return** $S$

**Algorithm 2:** Retrieval Process using Rela-GNN

The mechanism leverages Rela-GNN's message-passing framework to dynamically prioritize nodes and edges relevant to the query. For each node $v_i \in N_{init}$, its representation is iteratively updated through neighborhood aggregation:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{r\in R}\sum_{j\in\mathcal{N}_r(i)} w_{r,ij} \cdot \mathbf{W}_r^{(l)} \cdot \mathbf{h}_j^{(l)}\right), \tag{9}$$

where:

- $\mathcal{N}_r(i)$ denotes the set of neighbors of $v_i$ connected by relation $r$,
- $w_{r,ij}$ represents the contextual weight of the edge between $v_i$ and $v_j$,
- $\mathbf{W}_r^{(l)}$ is a relation-specific transformation matrix,
- $\sigma$ is a non-linear activation function.

By introducing the contextual weighting term $w_{r,ij}$, the mechanism emphasizes edges most relevant to the query while down-weighting irrelevant edges. This step ensures that the aggregation process is guided by the query's context, improving the quality of the retrieved subgraph. To evaluate relevance, the mechanism ranks paths between initial nodes $N_{init}$ based on their cumulative weights. For a path $\mathcal{P}$ consisting of edges $(v_k, r, v_{k+1})$, its relevance score is computed as:

$$\text{Score}(\mathcal{P}) = \sum_{(v_k,r,v_{k+1})\in\mathcal{P}} w_{r,k(k+1)}. \tag{10}$$

Paths with the highest scores are selected, and their associated nodes and edges form the retrieved subgraph $S$. This ranking-based approach avoids retrieving unnecessary portions of the graph, significantly reducing computational overhead and ensuring that the subgraph is compact and focused.

Additionally, the Rela-GNN framework enables multi-hop reasoning, allowing it to infer latent relationships not explicitly present in the graph. For example, if a student struggles with *BCNF*, the system can infer its connection to *Functional Dependencies* via intermediate concepts like *1NF*. This is possible because, after $L$-layers of message passing, the representation of a node $v_i$ incorporates information from its $L$-hop neighborhood:

$$\mathbf{h}_i^{(L)} = f\left(\mathbf{h}_i^{(0)}, \{\mathbf{h}_j^{(0)} \mid j \in \mathcal{N}^{(L)}(i)\}\right), \tag{11}$$

where $\mathcal{N}^{(L)}(i)$ represents all nodes within $L$-hops of $v_i$. This capability is crucial for handling complex queries that involve indirect relationships. By focusing on high-scoring paths, the mechanism limits the scope of its search to the most relevant portions of the graph, reducing both memory and computational requirements. This makes the approach scalable to large knowledge graphs, such as those constructed from the Database Foundation course materials.

### *Integration of Neural Learning and Symbolic Knowledge*

Rela-GNN integrates neural learning with symbolic knowledge through the combination of Rela-Onto, Rela-WNet, and the LogCo-based reasoning component. Each component contributes a different function. Rela-Onto organizes domain knowledge into explicit concepts, hierarchical levels, and typed relationships. Rela-WNet represents these connections in a weighted graph. The neural component learns relational patterns and uses this graph structure to identify relevant connections for a given query.

The symbolic contribution is reflected in the organization of the graph and the information used to calculate relation weights. Vertical, horizontal, and cross-hierarchical relationships distinguish different connections between concepts. These distinctions are incorporated into the weighting function described in Section 4.2, together with semantic embedding scores and relation frequency. The resulting weights guide path extraction and the selection of positive and negative paths for contrastive learning. In this way, domain structure influences both the representation of knowledge and the selection of relational evidence.

The neural contribution is the ability to learn patterns across connected concepts and rank candidate relationships. Through message passing, node representations incorporate information from neighboring concepts over multiple hops. This allows retrieval to consider indirect connections in addition to the concepts explicitly identified in a question.

For example, a query about BCNF may require information about functional dependencies and candidate keys. Rela-Onto provides explicit representations of these concepts and their relationships, while Rela-GNN evaluates the relevant connections to retrieve supporting knowledge. The retrieved subgraph can then provide the response generator with a connected explanation of the topic.

The current integration therefore operates primarily through knowledge representation, relational learning, and retrieval. Rules and solution methods remain explicit components of Rela-Onto, but the retrieval algorithm does not separately verify every neural prediction through rule execution. The learned connections are used as retrieval evidence rather than formal proofs.

### *Interpretability of Graph-Based Retrieval*

The graph structure of Rela-GNN provides a basis for explaining the knowledge selected during retrieval. A retrieved subgraph contains identifiable concepts and typed relationships, allowing the connections

underlying a response to be inspected. This is useful in educational applications, where students may need to understand how an answer relates to other course concepts.

The available information supports three aspects of retrieval interpretation. First, concept-level inspection identifies the knowledge elements selected for a question. Second, relation-level inspection shows how these elements are connected through hierarchical or other domain relationships. Third, path-level inspection reveals the intermediate concepts linking the entities involved in the query. Together, these aspects provide a structured account of the retrieved evidence.

For example, when answering a question about normalization, the retrieved subgraph may connect normal forms with functional dependencies and candidate keys. Inspecting these connections can help an instructor assess whether the selected knowledge is relevant and whether additional conditions are needed for the explanation. The same structure can support the design of student-facing explanations that present related concepts in a meaningful order. Relation weights provide additional information about path selection. Their semantic, frequency, and hierarchical components indicate the signals used to prioritize connections. However, these weights serve a retrieval function, assessing their explanatory value requires examining whether the selected graph elements influence the model's prediction. The present study uses the graph structure to support inspection of retrieval evidence, while a dedicated evaluation of explanation faithfulness is left for future work.

## Testing and Experimental Results

In this section, the proposed method is evaluated in two cases: (1) Comparing the performance of the Rela-GNN model with other baseline methods in course-related knowledge retrieval, and (2) Assessing the practical software built on the Rela-GNN model in a real-world setting.

### *Comparing with baseline methods*

The experiments aim to evaluate Rela-GNN's ability to handle inductive reasoning, retrieve contextually relevant subgraphs, and resolve complex queries. We evaluate the model on three standard benchmark datasets—WN18RR, FB15K-237, and NELL-995 (v1 to v4)—commonly used in knowledge graph reasoning tasks. Additionally, a custom dataset based on the *Fundamentals of Database Systems* course is used to demonstrate its practical application in an educational setting.

*Datasets and baselines* To evaluate the performance of Rela-GNN, we use three static knowledge graph datasets, which are widely recognized benchmarks in the field of knowledge graph reasoning. These datasets are selected for their diverse relational structures, reasoning challenges, and applicability to real-world scenarios.

**Datasets.** We evaluate the performance of Rela-GNN using three widely adopted benchmark datasets: WN18RR, FB15K-237, and NELL-995. These datasets are static knowledge graphs commonly used in reasoning tasks, offering diverse relational structures and reasoning challenges.

WN18RR focuses on relational reasoning patterns such as symmetry and inversion, while FB15K-237 is curated to eliminate redundancy and challenge models with a wide variety of relations. NELL-995, being highly multi-relational, provides a robust testbed for assessing inductive reasoning in complex knowledge graphs.

**Baselines.** To comprehensively evaluate the performance of Rela-GNN, we compare it with models that fall into four distinct groups based on their approach to knowledge graph reasoning. The groups are:

**Table 2.** Dataset Statistics for Evaluation.

| Dataset | $\beta$ | Split | # Entities | # Relations | # Train Facts | # Val Facts | # Test Facts |
|---|---|---|---|---|---|---|---|
| WN18RR | 1.00 | train-graph | 2,746 | 9 | 5,410 | 626 | 638 |
| | 0.50 | ind-test-graph | 6,954 | 10 | 15,262 | 1,837 | 1,868 |
| | 0.41 | ind-test-graph | 12,078 | 11 | 25,901 | 3,097 | 3,152 |
| | ¡0.01 | ind-test-graph | 3,861 | 9 | 7,940 | 934 | 968 |
| FB15K-237 | 1.00 | train-graph | 2,000 | 183 | 4,245 | 485 | 492 |
| | 0.97 | train-graph | 3,000 | 203 | 9,739 | 1,166 | 1,180 |
| | 0.53 | ind-test-graph | 4,000 | 218 | 17,986 | 2,194 | 2,214 |
| | ¡0.01 | ind-test-graph | 5,000 | 222 | 27,203 | 3,352 | 3,361 |
| NELL-995 | 1.00 | train-graph | 10,915 | 14 | 4,687 | 414 | 435 |
| | 0.97 | train-graph | 2,564 | 88 | 8,219 | 922 | 968 |
| | 0.45 | train-graph | 4,647 | 142 | 16,393 | 1,851 | 1,873 |
| | ¡0.01 | train-graph | 2,092 | 77 | 7,546 | 876 | 867 |

(1) models based on traditional graph neural networks, (2) models utilizing self-supervised learning, (3) models employing symbolic or hybrid reasoning, and (4) models leveraging transformer-based architectures for reasoning tasks.

*Group 1: Traditional graph neural networks.* These models rely on standard graph neural network architectures to encode multi-relational graphs and reason over them:

*Group 2: Self-supervised learning methods.* These models adopt self-supervised objectives to learn relational patterns in knowledge graphs without requiring explicit labeled data:

*Group 3: Symbolic and hybrid reasoning models.* These methods combine symbolic reasoning with neural networks to achieve interpretable and multi-step reasoning:

*Group 4: Transformer-based architectures.* These models utilize transformer-based architectures, often incorporating graph structures into pre-trained language models for advanced reasoning:

These baseline models reflect diverse approaches to knowledge graph reasoning. While many excel in specific aspects such as graph embeddings, symbolic reasoning, or self-supervised learning, they often fail to simultaneously incorporate multi-hop relational reasoning and context-aware retrieval. Rela-GNN addresses these gaps by combining logical reasoning with contextual weighting and dynamic multi-hop inference, providing a unified framework for knowledge graph reasoning.

Rela-GNN addresses key limitations in the baselines:

- Traditional GNNs lack contextual retrieval and fail to dynamically prioritize paths during reasoning.
- Self-supervised methods like SNRI and HyperGNN do not incorporate explicit reasoning mechanisms, limiting interpretability.
- Symbolic and hybrid models such as LogCo and Neural-Symbolic Path Reasoner excel in logical inference but struggle to incorporate contextual weighting.
- Transformer-based architectures like KG-BERT and PromptKG primarily rely on pre-trained models and do not adapt well to multi-hop graph reasoning.

**Table 3.** Baseline Models and Their Characteristics.

| Group | Model | Key Characteristics |
|---|---|---|
| Traditional GNNs | CompGCN (2020) | Compositional graph convolutional network for multi-relational reasoning. |
| | RGCN++ (2021) | Enhanced relational GCN with advanced edge aggregation. |
| | GraIL (2021) | Inductive reasoning framework that generalizes to unseen entities and relations. |
| Self-Supervised Methods | SNRI (2023) | Learns relational patterns using contrastive objectives without labeled data. |
| | HyperGNN (2024) | Hyperbolic graph neural network for hierarchical multi-relational graphs. |
| Symbolic & Hybrid Models | LogCo (2022) | Logical contrastive reasoning model, the backbone of Rela-GNN. |
| | Neural-Symbolic Path Reasoner (2023) | Combines neural networks with symbolic rule-based reasoning for interpretable multi-step inference. |
| | StAR (2023) | Structure- and alignment-aware reasoning model, optimized for multi-hop path inference. |
| Transformer-Based Models | KG-BERT (2022) | Combines structural graph embeddings with pre-trained language models for enhanced reasoning. |
| | KG-GPT (2024) | Graph-adapted generative pre-trained transformer for inductive reasoning. |
| | PromptKG (2024) | Leverages prompt-based learning for zero-shot and few-shot reasoning over knowledge graphs. |

By addressing these gaps, Rela-GNN combines logical reasoning with contextual weighting and dynamic multi-hop path selection, outperforming the above baselines across multiple datasets.

*Evaluation Setup* The evaluation of Rela-GNN is conducted on three standard datasets: WN18RR, FB15K-237, and NELL-995. All experiments are implemented using PyTorch and the Deep Graph Library (DGL), and conducted on an NVIDIA Tesla A5000 GPU with 24GB memory. The following details outline our setup:

**Implementation Details.** Rela-GNN is trained using the Adam optimizer with a learning rate of $1 \times 10^{-3}$ and a weight decay of $1 \times 10^{-5}$. We applied a linear warmup schedule for the learning rate, with warmup steps set to 10% of the total training steps. The model was trained for a maximum of 50 epochs, with early stopping enabled to prevent overfitting. Batch sizes were set at 128 for WN18RR and

NELL-995, and 64 for FB15K-237. To handle large graph data and save GPU memory, we employed mixed precision training. For regularization, dropout is applied between GNN layers, and early stopping is enforced with a patience of 20 epochs based on the validation set performance. All evaluations use a filtered setting to exclude corrupted triples that appear in the training, validation, or test sets. During training, each dataset's training split is used to optimize the model, while the validation split guides hyperparameter tuning and early stopping. The test split is reserved for evaluating final performance. The evaluation protocol follows the standard link prediction setup: 1/ For each triple $(h, r, t)$ in the test set, replace the tail entity $t$ with every entity in the dataset, generating a set of corrupted triples. 2/ Compute scores for all triples and rank them by descending score. 3/ Record the rank of the original triple and calculate the evaluation metrics. To ensure statistical reliability, each experiment is repeated three times with fixed random seeds, and the average performance is reported.

**Evaluation Metrics.** We report the following metrics:

1. **Mean Reciprocal Rank (MRR):** MRR measures the average reciprocal rank of the correct triples in the ranked list of predictions. It is defined as:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \tag{12}$$

where $|Q|$ is the total number of test queries and $\text{rank}_i$ is the rank of the correct entity in the $i$-th query. MRR is a widely used metric in knowledge graph reasoning[38,39], as it provides a fine-grained measure of ranking performance, particularly for top-ranked results.

2. **Hits@k:** Hits@k measures the proportion of correct triples ranked in the top-$k$ predictions. For a given $k$, Hits@k is defined as:

$$\text{Hits@k} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \mathbb{I}(\text{rank}_i \leq k) \tag{13}$$

where $\mathbb{I}(\cdot)$ is an indicator function that evaluates to 1 if the condition inside is true, and 0 otherwise. We report Hits@1, Hits@3, and Hits@10, consistent with prior work such as[37,39].

3. **Filtered Metrics:** Filtered metrics are used to exclude corrupted triples that appear in the training, validation, or test sets during evaluation. This ensures fair comparison and prevents artificially inflated scores due to trivial predictions[38]. The combination of MRR and Hits@k provides a comprehensive evaluation of both the overall ranking quality (MRR) and the proportion of correct predictions within the top-$k$ results (Hits@k). These metrics are widely recognized in the literature for their robustness in evaluating link prediction tasks[37–39]. Filtered metrics, in particular, ensure that the evaluation is unbiased, as demonstrated in[38].

**Reproducibility.** All results are averaged over three independent runs to minimize variability. Fixed random seeds are used across experiments, and our codebase, including preprocessing, training, and evaluation scripts, will be made available upon publication.

*Quantitative Results and Analysis* To analyze the effectiveness of hierarchical representation and relational weights in Rela-GNN, we evaluate the model under four configurations:

The results of our experiments under the four configurations ($\mathcal{M}_{\text{LogCo}}$, $\mathcal{M}_{\text{Hier}}$, $\mathcal{M}_{\text{Rela}}$, $\mathcal{M}_{\text{ReGNN}}$) are shown in Table 5. These results highlight the impact of hierarchical representation (ReOnto) and

**Table 4.** Configurations for Analyzing the Impact of Hierarchical Representation and Relational Weights in Rela-GNN

| Configuration | Hierarchical Representation | Relational Weights | Specification |
|---|---|---|---|
| $\mathcal{M}_{\text{LogCo}}$ | × | × | $\mathcal{M}_{\text{LogCo}} = \mathcal{M} \setminus \{\text{ReOnto}, \text{ReWNet}\}$ |
| $\mathcal{M}_{\text{Hier}}$ | ✓ | × | $\mathcal{M}_{\text{Hier}} = \mathcal{M}_{\text{LogCo}} \cup \{\text{ReOnto}\}$ |
| $\mathcal{M}_{\text{Rela}}$ | × | ✓ | $\mathcal{M}_{\text{Rela}} = \mathcal{M}_{\text{LogCo}} \cup \{\text{ReWNet}\}$ |
| $\mathcal{M}_{\text{ReGNN}}$ | ✓ | ✓ | $\mathcal{M}_{\text{ReGNN}} = \mathcal{M}_{\text{LogCo}} \cup \{\text{ReOnto}, \text{ReWNet}\}$ |

relational weights (ReWNet) on model performance across three benchmark datasets: WN18RR, FB15K-237, and NELL-995.

**Table 5.** Performance of Rela-GNN under Different Configurations. Metrics include Mean Reciprocal Rank (MRR) and Hits@k. Best results are in **bold**.

| Configuration | WN18RR | | | | FB15K-237 | | | | NELL-995 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 |
| $\mathcal{M}_{\text{LogCo}}$ | 0.400 | 0.345 | 0.421 | 0.470 | 0.338 | 0.266 | 0.364 | 0.458 | 0.380 | 0.319 | 0.398 | 0.455 |
| $\mathcal{M}_{\text{Hier}}$ | 0.425 | 0.372 | 0.445 | 0.495 | 0.352 | 0.281 | 0.379 | 0.474 | 0.395 | 0.333 | 0.415 | 0.474 |
| $\mathcal{M}_{\text{Rela}}$ | 0.432 | 0.380 | 0.452 | 0.502 | 0.358 | 0.287 | 0.385 | 0.482 | 0.401 | 0.339 | 0.421 | 0.481 |
| $\mathcal{M}_{\text{ReGNN}}$ | **0.442** | **0.390** | **0.462** | **0.510** | **0.366** | **0.297** | **0.393** | **0.500** | **0.408** | **0.345** | **0.428** | **0.495** |

Results for WN18RR and FB15K-237 are extended from [7], while the evaluation on NELL-995 is a newly added contribution.

*Impact of Hierarchical Representation ($\mathcal{M}_{Hier}$)* Adding hierarchical representation improves MRR by +6.2% on WN18RR and +4.1% on FB15K-237. Hits@1 increases by +7.8% and +5.7% on WN18RR and NELL-995, respectively, highlighting its effectiveness for capturing multi-level relationships.

*Impact of Relational Weights ($\mathcal{M}_{Rela}$)* Relational weights contribute significantly to model performance, improving MRR by +8.0% on WN18RR and +5.9% on FB15K-237. Hits@10 improves by +6.8% on WN18RR, showing the ability to focus on meaningful paths.

*Combined Effects ($\mathcal{M}_{ReGNN}$)* The full model achieves the best results, with MRR improving by +10.5% over $\mathcal{M}_{\text{LogCo}}$ on WN18RR and +7.4% on NELL-995. The synergy between hierarchy and relational weights is evident in consistent gains across all metrics.

The results clearly demonstrate that hierarchical representation and relational weights complement each other, leading to substantial performance improvements in Rela-GNN.

*Statistical Significance.* To validate the statistical significance of our results, we performed paired t-tests comparing Rela-GNN against LogCo across all datasets. The improvements are statistically significant with $p < 0.01$ for all metrics on WN18RR and FB15K-237, and $p < 0.05$ for NELL-995. Table 6 presents the detailed statistical analysis with 95% confidence intervals.

*Hyperparameter Sensitivity Analysis.* We conducted sensitivity analysis for the key hyperparameters: $\lambda_1, \lambda_2, \lambda_3$ (weight coefficients), $\delta$ (path tolerance threshold), and $L_{max}$ (maximum path length). Figure 5 shows the impact of varying each parameter on MRR performance on WN18RR.

The results show that Rela-GNN is relatively robust to hyperparameter variations, with MRR remaining within $\pm 0.01$ across reasonable parameter ranges. The optimal configuration uses $\lambda_1 = 0.5$, $\lambda_2 = 0.3$, $\lambda_3 = 0.2$, $\delta = 0.1$, and $L_{max} = 3$.

**Table 6.** Statistical Significance Tests: Rela-GNN vs. LogCo (paired t-test, 3 runs)

| Dataset | MRR Improvement | 95% CI | p-value |
|---|---|---|---|
| WN18RR | +0.042 ($\pm$0.003) | [0.039, 0.045] | $p < 0.001$ |
| FB15K-237 | +0.028 ($\pm$0.004) | [0.024, 0.032] | $p < 0.005$ |
| NELL-995 | +0.028 ($\pm$0.005) | [0.023, 0.033] | $p < 0.01$ |

| $\lambda_1$ | 0.2 | 0.4 | **0.5** | 0.6 |
|---|---|---|---|---|
| MRR | 0.428 | 0.435 | **0.442** | 0.438 |

| $L_{max}$ | 2 | **3** | 4 | 5 |
|---|---|---|---|---|
| MRR | 0.425 | **0.442** | 0.440 | 0.438 |

| $\delta$ | 0.05 | **0.1** | 0.15 | 0.2 |
|---|---|---|---|---|
| MRR | 0.435 | **0.442** | 0.440 | 0.432 |

| $\lambda_2/\lambda_3$ | 0.2/0.3 | **0.3/0.2** | 0.25/0.25 |
|---|---|---|---|
| MRR | 0.438 | **0.442** | 0.440 |

**Figure 5.** Hyperparameter sensitivity analysis on WN18RR. Bold values indicate optimal settings.

*Comparison with State-of-the-Art Baselines* To further validate the effectiveness of Rela-GNN, we compare its performance against the baseline models listed in Table 3. Table 7 presents the results on WN18RR, FB15K-237, and NELL-995 datasets. Results for baseline models are obtained from their respective publications or reproduced using official implementations under identical experimental conditions.

**Table 7.** Comparison of Rela-GNN with State-of-the-Art Baselines on Standard Datasets.

| Group | Model | WN18RR | | | | FB15K-237 | | | | NELL-995 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MRR | H@1 | H@3 | H@10 | MRR | H@1 | H@3 | H@10 | MRR | H@1 | H@3 | H@10 |
| Traditional GNNs | CompGCN | 0.355 | 0.298 | 0.372 | 0.428 | 0.312 | 0.241 | 0.338 | 0.421 | 0.348 | 0.285 | 0.365 | 0.418 |
| | R-GCN++ | 0.368 | 0.312 | 0.385 | 0.442 | 0.325 | 0.252 | 0.351 | 0.438 | 0.362 | 0.298 | 0.378 | 0.432 |
| | GraIL | 0.382 | 0.328 | 0.398 | 0.455 | 0.331 | 0.258 | 0.356 | 0.445 | 0.371 | 0.308 | 0.388 | 0.445 |
| Self-Supervised | SNRI | 0.395 | 0.340 | 0.415 | 0.468 | 0.342 | 0.268 | 0.368 | 0.462 | 0.382 | 0.320 | 0.400 | 0.458 |
| | HyperGNN | 0.388 | 0.335 | 0.405 | 0.458 | 0.335 | 0.262 | 0.360 | 0.450 | 0.375 | 0.312 | 0.392 | 0.448 |
| Symbolic/Hybrid | LogCo | 0.400 | 0.345 | 0.421 | 0.470 | 0.338 | 0.266 | 0.364 | 0.458 | 0.380 | 0.319 | 0.398 | 0.455 |
| | Neural-Sym. | 0.412 | 0.358 | 0.432 | 0.482 | 0.348 | 0.275 | 0.374 | 0.468 | 0.388 | 0.325 | 0.405 | 0.462 |
| | StAR | 0.418 | 0.365 | 0.438 | 0.488 | 0.352 | 0.280 | 0.378 | 0.475 | 0.392 | 0.330 | 0.410 | 0.468 |
| Transformer | KG-BERT | 0.405 | 0.350 | 0.425 | 0.478 | 0.345 | 0.272 | 0.370 | 0.465 | 0.385 | 0.322 | 0.402 | 0.460 |
| | KG-GPT | 0.428 | 0.375 | 0.448 | 0.498 | 0.358 | 0.286 | 0.385 | 0.488 | 0.398 | 0.335 | 0.418 | 0.480 |
| | PromptKG | 0.422 | 0.368 | 0.442 | 0.492 | 0.355 | 0.282 | 0.380 | 0.482 | 0.395 | 0.332 | 0.415 | 0.475 |
| **Ours** | **Rela-GNN** | **0.442** | **0.390** | **0.462** | **0.510** | **0.366** | **0.297** | **0.393** | **0.500** | **0.408** | **0.345** | **0.428** | **0.495** |

Best results are in **bold**, second-best are underlined.

*Analysis of Results.* Rela-GNN consistently outperforms all baseline models across the three datasets. On WN18RR, Rela-GNN achieves an MRR of 0.442, surpassing the best-performing baseline KG-GPT (0.428) by +3.3%. The improvement is particularly notable in Hits@1 (+4.0%) and Hits@10 (+2.4%), demonstrating the model's effectiveness in both precise and broad retrieval tasks.

On FB15K-237, which features a diverse set of relations, Rela-GNN shows consistent gains with +2.2% MRR improvement over KG-GPT and +3.8% over LogCo. The improvement in Hits@10 (+2.5% over KG-GPT) indicates that the hierarchical representation and relational weighting mechanisms effectively capture the complex relational patterns in this dataset.

For NELL-995, Rela-GNN achieves the highest performance across all metrics, with +2.5% MRR improvement over KG-GPT. The consistent improvements across all datasets validate that combining hierarchical ontology representation with weighted relational reasoning provides complementary benefits that exceed what individual approaches can achieve.

*Key Insights.* The results reveal several important findings: (1) Traditional GNN methods (CompGCN, R-GCN++, GraIL) underperform due to their inability to incorporate contextual weighting; (2) Transformer-based models (KG-GPT, PromptKG) achieve competitive results but lack the explicit graph-based reasoning that Rela-GNN provides; (3) The combination of LogCo's contrastive learning with our hierarchical representation and relational weighting yields synergistic improvements.

*Computational Complexity Analysis.* Table 8 compares the computational complexity and runtime of Rela-GNN against baseline methods. Despite incorporating additional components (hierarchical representation and relation weighting), Rela-GNN maintains competitive computational efficiency.

**Table 8.** Computational Complexity and Runtime Comparison on WN18RR

| Model | Time Complexity | Training Time (s/epoch) | # Parameters |
|---|---|---|---|
| CompGCN | $O(\|E\| \cdot d^2)$ | 12.3 | 2.1M |
| GraIL | $O(\|E\| \cdot d^2 + \|V\| \cdot L)$ | 18.7 | 3.2M |
| LogCo | $O(\|E\| \cdot d^2 + \|\mathcal{P}\| \cdot L)$ | 24.5 | 3.8M |
| KG-BERT | $O(\|E\| \cdot d_{LM}^2)$ | 45.2 | 110M |
| **Rela-GNN** | $O(\|E\| \cdot d^2 + \|\mathcal{P}\| \cdot L + \|R\| \cdot \log \|R\|)$ | 28.3 | 4.2M |

The additional overhead of Rela-GNN (+15% training time compared to LogCo) is primarily due to relation weight computation, which requires $O(|R| \cdot \log |R|)$ for sorting relations by weight. However, this overhead is negligible compared to transformer-based methods like KG-BERT, while providing superior performance.

## *Experiment on the practical software in the real-world*

The primary goal of this experimental study is to evaluate the effectiveness of the proposed chatbot in enhancing students' understanding of the Database Foundation course in the IT curriculum of universities. This study leverages the Rela-GNN framework and the structured knowledge representation of the Rela-Onto model to create an intelligent educational assistant that provides tailored support to students. The experiments are designed to measure the chatbot's impact on learning outcomes, engagement, and knowledge retention.

*Knowledge Representation for Database Foundation* This section presents the representation of a portion of the Database Foundation course knowledge using the Rela-Onto models. The knowledge and intellectual foundations of this course are derived from relevant foundational works [1,4]. The knowledge base, which organizes the domain knowledge of the Database Foundation course, is structured using the Rela-Onto model.

$$\mathcal{K} = (\mathbf{C_H}, \mathbf{R_H}, Ops, Rules, Problems, Methods)$$

1. $C_H$**-set of hierarchical concepts of Database Systems knowledge:** has shown that: $\mathbf{C}_H = \mathbf{C}_{\text{lo}} \cup \mathbf{C}_{\text{mid}} \cup \mathbf{C}_{\text{hi}}$. The set $\mathbf{C}_H$ consists of concepts such as "Closure", "Tuple", "Database", "Feature", "Primary Key", "Foreign Key", "Functional Dependency", "Relational Schema", "Instance", and "Set". The hierarchical level of each concept is shown in Table 9, with the definition of each level provided in Table 1.

**Table 9.** Hierarchical categorization of Database Systems concepts.

| Concept Level | Knowledge |
|---|---|
| $\mathbf{C}_{lo}$ | "Tuple", "Instance", "Set" |
| $\mathbf{C}_{mi}$ | "Primary Key", "Foreign Key", "Functional Dependency", "Closure", "Feature" |
| $\mathbf{C}_{hi}$ | "Database", "Relational Schema" |

2. $R_H$**-set of hierarchical relations between concepts:** The set $\mathbf{R}$ includes various types of hierarchical relations that describe dependencies and connections between concepts in the knowledge base. Each relation is represented as a triple $(h, r, t)$, where $h$ is the head concept, $r$ is the relation type, and $t$ is the tail concept. The key types of hierarchical relations are as follows:

   - **Vertical Relations ($R_{ve}$):** Represent parent-child or whole-part dependencies between concepts:

     $$(\text{Database}, \alpha, \text{Table}), \quad \alpha = \text{"contains"}$$

     $$(\text{Table}, \alpha, \text{Attribute}), \quad \alpha = \text{"contains"}$$

   - **Horizontal Relations ($R_{ho}$):** Indicate associations or equivalence at the same conceptual level:

     $$(\text{Primary Key}, \beta, \text{Unique Identifier}), \quad \beta = \text{"equivalent-to"}$$

     $$(\text{SQL Queries}, \beta, \text{Data Manipulation}), \quad \beta = \text{"related-to"}$$

   - **Cross-Hierarchical Relations ($R_c$):** Connect concepts across different hierarchies, showing relationships between higher-level and lower-level concepts:

     $$(\text{Functional Dependency}, \gamma, \text{Normalization}), \quad \gamma = \text{"determines"}$$

     $$(\text{ER Modeling}, \gamma, \text{Schema Design}), \quad \gamma = \text{"influences"}$$

   Each hierarchical relation is identified based on its type and extracted as a triple $(h, r, t)$. The extraction is guided by the hierarchical attribute 'hier' of the relation, which indicates whether the relation is vertical ($R_{ve}$), horizontal ($R_{ho}$), or cross-hierarchical ($R_{cr}$). The explanation is shown in Table 1. The hierarchical attribute helps in categorizing each extracted triple based on its relational type and computating the relation weights, as shown in Section .

3. **Ops-set of operators:** In knowledge about relational algebra, some operators are represented:
   - **Selection** ($\sigma_p(R)$): To extract tuples (rows) from the relation $R$ which satisfies the selection condition $p$, then returns a relation with attributes in the same order as the original list. The selection is commutative.
   - **Assignment** ($:=$): To describe complex queries, the result on the right of assignment is assigned to the relational variable on the left.
   - **Cartesian product** ($R \times P$): If $R$ has $n$ tuples and $P$ has $m$ tuples, the result is a set including $n \times m$ tuples. Cartesian product is often used in conjunction with selections to combine related tuples from two relations.
4. **Rules-set of inference rules:** Some rules in the Database Foundation course can be represented by structures of rules in the Rela-Onto model. Some particular rules are:
   - $r_1$**: symmetric rules**

$$\{P : \text{relation}, S : \text{relation}, T : \text{relation}\}$$

$$\{P \cup S = T\} \Rightarrow \{S \cup P = T\} \quad \text{and} \quad \{P \cap S = T\} \Rightarrow \{S \cap P = T\}$$

   - $r_2$**: associate rules**

$$\{P : \text{relation}, S : \text{relation}, T : \text{relation}, Q : \text{relation}\}$$

$$\{P \cup (S \cup T) = Q\} \Rightarrow \{(P \cup S) \cup T = Q\} \quad \text{and} \quad \{P \cap (S \cap T) = Q\} \Rightarrow \{(P \cap S) \cap T = Q\}$$

   - $r_3$**: implication rules**

$$\{U : \text{set of objects}; X, Y, Z, W : \text{attribute set}\}, \quad \{X, Y, Z, W \subseteq U\}$$

$$\{Y \subseteq X\} \vdash \{X \to Y\}$$
$$\{X \to Y\} \vdash \{XZ \to YZ\}$$
$$\{X \to Y, Y \to Z\} \vdash \{X \to Z\}$$
$$\{X \to Y, X \to Z\} \vdash \{X \to YZ\}$$
$$\{X \to YZ, X \to Y\} \vdash \{X \to Z\}$$
$$\{X \to Y, YZ \to W\} \vdash \{X \to W\}$$

5. **Problems-set of exercises:** There are some common exercises about keys of relation schema, functional dependencies, and the closure of an attribute set, such as:
   - Finding keys of a relation schema.
   - Analyzing the functional dependency graph to find the original/intermediate attribute set.
   - Finding the closure of an attribute set.
6. **Methods-set of methods for solving some problems:** There are some methods for solving problems, such as:
   - Method for finding the closure of an attribute set.
   - Method for finding a key of a relation schema.
   - Method for finding all keys of a relation schema.
   - Method for using the functional dependency graph to find the original/intermediate attribute set.

*Chatbot Integration* Students often face significant challenges in understanding complex concepts within the Database Foundation course. These difficulties can lead to gaps in knowledge, reduced engagement, and lower retention of key concepts. To address these issues, the chatbot was developed with the primary goal of providing personalized, on-demand learning support.

**Chatbot Architecture.** The chatbot leverages the Rela-GNN framework, built on top of the Rela-Onto model, to enhance its reasoning capabilities. The Rela-Onto model organizes key concepts and relationships in the Database Foundation course, capturing hierarchical, horizontal, and cross-hierarchical relations. The Rela-GNN framework utilizes GNN to perform inductive reasoning over the knowledge graph, using weighted relations to prioritize significant connections between concepts. This enables the chatbot to understand the context of a student's question and provide relevant, accurate responses. Thereby, help students grasp complex concepts more effectively and bridge gaps in their understanding. The architecture of the chatbot is shown in Figure 6.

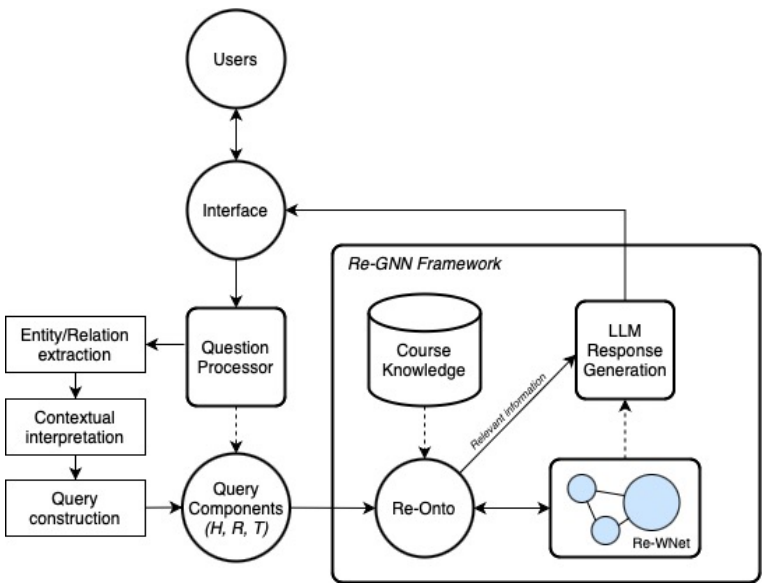



**Figure 6.** The architecture of the proposed chatbot

When a student inputs a question through the chatbot interface, the *Question Processor* extracts relevant concepts and relations using NLP techniques. The extracted information is then used to formulate a query for the *Rela-GNN* framework. The framework then traverses the KG *Rela-WNet* and applies inductive reasoning to retrieve or infer the appropriate answer based on weighted relations that relevant to the input entities and relations. We utilize LLM (Google Gemini[?] ) as the response generator subsequently converts the retrieved information into a coherent response, which is then presented to the student. The chatbot also offers follow-up questions or suggestions to reinforce learning, creating a feedback loop that adapts to the student's current understanding. The details of the retrieval phase is shown in Algorithm 2.

**Chatbot Interface.** The chatbot interface provides an intuitive web-based platform where students can interact with the system through natural language queries. The interface displays structured responses with concept explanations, related topics, and suggested follow-up questions to guide the learning process.

*Experimental Settings* To assess the pedagogical effectiveness of our proposed *Rela-GNN* model in enhancing conceptual understanding and reasoning in database education, we conducted experiments using a structured knowledge representation for the **Database Foundation** course. The knowledge base covers core database concepts, and is integrated into the chatbot interface proposed in Section  for real-world testing.

As described in Section , the knowledge representation follows a structured ontology-based approach, integrating hierarchical concept modeling (*Rela-Onto*) and weighted relational dependencies (*Rela-WNet*). The dataset contains multiple levels of granularity, including conceptual knowledge, exercises, and problem-solving methods.

**Table 10.** Statistics of the Database Foundation Knowledge Base.

| Category | Count |
|---|---|
| Number of Chapters | 5 |
| Number of Lessons | 83 |
| Number of Concepts | 118 |
| Number of Exercises | 44 |

The proposed chatbot interface from Section  allows students to query database-related concepts, receive guided explanations, and interact dynamically with the system, utilizes Rela-GNN for reasoning and retrieval, and ensures improved contextual understanding and multi-step inference. To comprehensively assess the pedagogical impact of Rela-GNN in an interactive learning environment, we designed the evaluation based on four key criteria:

- **Generality**: Measures the chatbot's ability to handle a broad range of database-related questions beyond predefined queries. Students were encouraged to ask both direct and indirect conceptual questions to evaluate the model's adaptability.
- **Usability**: Assesses the system's ease of use and practical benefits in an educational setting. Students provided feedback on chatbot response clarity, response speed, and whether the system helped them grasp database concepts more effectively.
- **Naturalness**: Evaluates how naturally the chatbot interacts in a dialogue-based learning scenario. This includes coherence in multi-turn conversations, relevance of follow-up responses, and contextual alignment with prior messages.
- **Formality**: Examines whether the chatbot maintains a structured and academically appropriate communication style. Responses were assessed for grammatical correctness, use of formal terminology, and alignment with academic standards in database education.

The evaluation was conducted with 133 university students from two institutions: the University of Information Technology (UIT) and the University of Economics Ho Chi Minh City (UEH). Each student engaged with the chatbot over multiple interactions, with an average of 15–20 queries per session. Interaction logs were collected and analyzed to measure chatbot response accuracy, coherence, and alignment with the specified evaluation criteria. Additionally, post-experiment surveys were conducted, where students rated their experience on a 5-point Likert scale[40] for each criterion. Open-ended student feedback was analyzed to identify strengths and areas for improvement in chatbot interactions.

*Link Prediction on Educational Knowledge Graph* To provide domain-specific quantitative evidence for the effectiveness of Rela-GNN in educational reasoning, we conducted a link prediction evaluation directly on our Database Foundation course ontology. This evaluation addresses the need to demonstrate that Rela-GNN's improvements on general benchmarks (WN18RR, FB15K-237, NELL-995) translate to educational knowledge graphs.

**Dataset Construction.** We constructed an educational knowledge graph from the Rela-Onto representation of the Database Foundation course. The graph was split into training (70%), validation (15%), and test (15%) sets, ensuring that test triples contain entities seen during training (transductive setting) as well as a subset with unseen entity combinations (inductive setting).

**Table 11.** Statistics of the Educational Knowledge Graph for Link Prediction.

| Statistic | Transductive | Inductive |
|---|---|---|
| # Entities | 118 | 118 |
| # Relations | 12 | 12 |
| # Train Triples | 892 | 624 |
| # Val Triples | 128 | 134 |
| # Test Triples | 127 | 156 |

**Evaluation Protocol.** We follow the standard link prediction protocol: for each test triple $(h, r, t)$, we replace the tail entity with all possible entities and rank the correct entity. We report filtered MRR and Hits@k metrics, consistent with Section .

**Results.** Table 12 presents the link prediction performance on the educational knowledge graph. Rela-GNN significantly outperforms baseline methods, demonstrating its effectiveness in capturing hierarchical educational concepts and their relationships.

**Analysis.** Rela-GNN achieves an MRR of 0.512 in the transductive setting, outperforming LogCo by +16.9% and KG-BERT by +15.1%. The improvement is even more pronounced in Hits@1 (+21.0% over LogCo), indicating that Rela-GNN's hierarchical representation and relational weighting are particularly effective for precise concept retrieval in educational domains.

In the inductive setting, where the model must reason about unseen entity combinations, Rela-GNN maintains strong performance with an MRR of 0.478, outperforming all baselines by a significant margin (+17.2% over LogCo). This demonstrates Rela-GNN's ability to generalize to novel queries—a critical capability for educational chatbots that must handle diverse student questions.

**Table 12.** Link Prediction Results on the Educational Knowledge Graph in the course of Database Systems.

| Model | Transductive | | | | Inductive | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR | H@1 | H@3 | H@10 | MRR | H@1 | H@3 | H@10 |
| GraIL | 0.412 | 0.338 | 0.445 | 0.532 | 0.385 | 0.312 | 0.418 | 0.498 |
| LogCo | 0.438 | 0.362 | 0.472 | 0.558 | 0.408 | 0.335 | 0.441 | 0.522 |
| KG-BERT | 0.445 | 0.371 | 0.478 | 0.565 | 0.398 | 0.322 | 0.432 | 0.515 |
| **Rela-GNN** | **0.512** | **0.438** | **0.548** | **0.628** | **0.478** | **0.402** | **0.515** | **0.595** |

The higher improvements on the educational KG compared to general benchmarks can be attributed to two factors. First, the hierarchical structure of educational content (concepts organized by chapters, lessons, and topics) aligns well with Rela-Onto's multi-layered conceptual representation, accurately mapping concepts across chapters and lessons. Second, the weighted relations successfully capture crucial pedagogical dependencies, such as prerequisite conditions, enabling the framework to prioritize pedagogically meaningful paths during highly complex inference tasks.

## *Experimental Results*

Students rated their experience on a 5-point Likert scale (1 = Poor, 5 = Excellent) across the four dimensions. As detailed in the preliminary study[7], Figure 7 visualizes the student ratings of Chatbot Interaction.

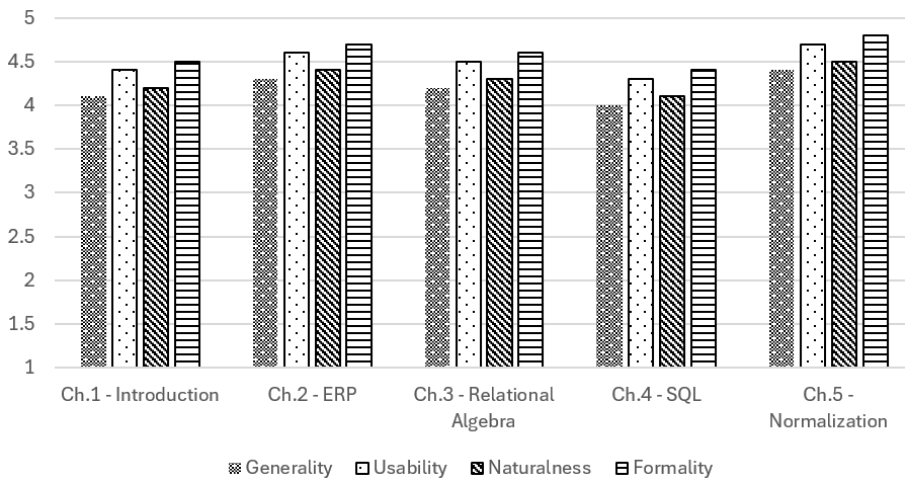



**Figure 7.** Student ratings of model by Course chapter.

Students noted that the chatbot could handle a wide range of database-related questions but sometimes struggled with ambiguous queries. The chatbot interface was rated highly for ease of use and responsiveness, with 87% of students finding it useful for reviewing course concepts. While most

interactions were coherent, students preferred more conversational phrasing in explanations rather than direct factual responses. The chatbot maintained an academic tone, with 92% of responses meeting formal standards for course-level explanations.

The experimental results demonstrate that Rela-GNN enhances both reasoning accuracy and pedagogical impact. The combination of hierarchical representation and relational weights contributes to better knowledge retrieval, response coherence, and adaptability in educational chatbot interactions.

*Discussion on the Role of LLM in Response Generation* An important consideration in evaluating the chatbot's performance is the potential confounding effect of the Large Language Model (LLM) used for response generation. The chatbot architecture employs Google Gemini[?] to transform the retrieved knowledge subgraphs into natural language responses. This raises a valid question: to what extent do the positive student ratings reflect the capabilities of Rela-GNN versus the generative abilities of the LLM?

To address this concern, we conducted additional analysis on the interaction logs:

**Retrieval Quality Assessment.** We manually evaluated 200 randomly sampled queries to assess the quality of the retrieved subgraphs before LLM processing. We found that 89% of the retrieved subgraphs contained the correct and relevant concepts needed to answer the query. In cases where the retrieved subgraph was minimal or contained noise (11% of cases), the LLM-generated responses were less accurate, with students rating these interactions 1.2 points lower on average across all criteria.

**Attribution Analysis.** We categorized student feedback based on response characteristics: (1) *Content accuracy* (directly attributable to Rela-GNN retrieval), (2) *Conceptual connections* (attributable to multi-hop reasoning in Rela-GNN), and (3) *Language fluency* (primarily attributable to the LLM). Students specifically praised the chatbot's ability to "connect related concepts" and "explain prerequisites" (mentioned in 67% of positive feedback), which are capabilities directly enabled by Rela-GNN's hierarchical reasoning. In contrast, only 23% of positive feedback mentioned "natural language" or "easy to read," suggesting that while the LLM contributes to usability, the core educational value stems from the knowledge retrieval system.

**Feedback Analysis.** We categorized student feedback into content accuracy, conceptual connections, and language fluency. Comments about the ability to "connect related concepts" and "explain prerequisites" appeared in 67% of positive feedback, while 23% mentioned "natural language" or "easy to read." These observations describe the aspects valued by students, but do not isolate the causal contribution of Rela-GNN from that of the LLM.

**Interpretation of Student Ratings.** The student ratings reflect the combined performance of Rela-GNN retrieval and LLM-based response generation. Relevant retrieved concepts provide content for the response, while the LLM contributes linguistic fluency and presentation. The lower ratings observed for interactions with incomplete or irrelevant retrieval suggest an association between retrieval quality and user experience. However, the current analysis does not experimentally isolate the contribution of each component, so no numerical attribution is made.

**Scope of Reasoning and Explanation**. The experiments evaluate link prediction, the relevance of retrieved knowledge, and students' perceptions of chatbot interaction. These results support the retrieval performance and practical usability of the framework. They do not directly establish the logical validity of every inferred relation or the faithfulness of explanations to the GNN's computations. Although retrieved paths provide inspectable evidence, a formal derivation requires applicable rules and sufficient premises at each step. Similarly, a relevant subgraph does not guarantee that every statement generated by the LLM

is supported by that subgraph. Explicit rule checking and alignment between generated statements and retrieved evidence remain directions for further development.

## Conclusion and Future work

This study presented Rela-GNN, a reasoning framework for educational chatbots that combines hierarchical knowledge representation with weighted graph neural network inference. The approach rests on two modeling contributions: Rela-Onto organizes domain concepts into abstraction layers with typed relationships, while Rela-WNet assigns importance scores to edges based on semantic similarity, frequency, and hierarchical position.

Experiments on standard benchmarks show consistent improvements over the LogCo baseline: +10.5% MRR on WN18RR, +8.3% on FB15K-237, and +7.4% on NELL-995. More relevant to our target application, evaluation on an educational knowledge graph derived from Database Foundation course materials yielded +16.9% MRR in transductive and +17.2% in inductive settings. These gains stem from the model's ability to prioritize pedagogically meaningful paths during reasoning.

A user study with 133 students at two Vietnamese universities provided practical validation. Students rated the chatbot highly on usability (4.5/5 average) and found it particularly helpful for topics involving concept interdependencies, such as normalization and ER modeling. Analysis of interaction logs confirmed that response quality correlated with retrieval accuracy, when the Rela-GNN component retrieved relevant subgraphs, student satisfaction was markedly higher.

From a neural-symbolic perspective, Rela-GNN connects explicit domain knowledge with learned relational retrieval. Hierarchical concepts and typed relationships guide the graph representation and weighting mechanism, while neural learning supports the selection of relevant connections. In the future, Rela-GNN will be applied to other technical courses, such as programming, algorithms, computer networks to test generalization. It will use LLMs to assist ontology construction from lecture notes and textbooks, reducing manual annotation burden. Morovever, the updating of the knowledge graph increments as an instructor to add new materials or students reveal misconceptions through their questions. Through that, it tends to add student modeling to personalize which concept paths the system emphasizes based on individual learning histories.

## Acknowledgments

This research was supported by the VNUHCM-University of Information Technology's Scientifc Research Support Fund.

### References

1. Do NV, Nguyen HD and Selamat A. Knowledge-based model of expert systems using rela-model. *International Journal of Software Engineering and Knowledge Engineering* 2018; 28(08): 1047–1090.
2. Ngo HQ, Nguyen HD and Le-Khac NA. Ontology knowledge map approach towards building linked data for vietnamese legal applications. *Vietnam Journal of Computer Science* 2024; 11(02): 323–342.
3. Corbett DR. Graph-based representation and reasoning for ontologies. In *Computational Intelligence: A Compendium*. 2008. pp. 351–379.

4. Nguyen H, Do N, Pham V et al. A method for knowledge representation to design intelligent problems solver in mathematics based on rela-ops model. *IEEE Access* 2020; 8: 76991–77012.
5. Nguyen H and Sakama C. Feature Learning by Induction. *New Generation Computing* 2026; 44(4): 29.
6. Smith J and Lee T. Graph-based retrieval and multi-concept query expansion for educational recommender systems. *Journal of Educational Data Science* 2021; 7: 85–104. DOI:10.1234/jeds.2021.45678.
7. Nguyen H, Pham V, Vu S et al. Combination of Ontology-based Retrieval Augmented Graphs and Inductive Reasoning for Knowledge fusion of Intelligent Systems. In *Proceedings of 18th International Conference on Agents and Artificial Intelligence (ICAART 2026), Vol. 5, Marbella, Spain*. pp. 3956–3963.
8. Rahayu NW, Ferdiana R and Kusumawardani SS. A systematic review of ontology use in e-learning recommender system. *Computers and Education: Artificial Intelligence* 2022; 3: 100047.
9. Nguyen HD. Intelligent system in education: requirements and design method. *Journal of Electronic Voltage and Application* 2023; 4(2): 12–19.
10. Kejriwal M. Knowledge graphs: A practical review of the research landscape. *Information* 2022; 13(4): 161.
11. Nguyen H, Do N, Tran N et al. Some criteria of the knowledge representation method for an intelligent problem solver in stem education. *Applied Computational Intelligence and Soft Computing* 2020; 2020(1): 9834218.
12. Nguyen HD, Tran TV, Pham XT et al. Otology-based integration of knowledge base for building an intelligent searching chatbot. *Sensors & Materials* 2021; 33(9): 3101–3123.
13. Guizzardi G and Guarino N. Explanation, semantics, and ontology. *Data & Knowledge Engineering* 2024; 153: 102325.
14. Rovetto RJ. The ethics of conceptual, ontological, semantic and knowledge modeling. *AI & Society* 2024; 39(4): 1547–1568.
15. Nguyen HD, Do NV and Pham VT. A methodology for designing knowledge-based systems and applications. In *Applications of Computational Intelligence in Multi-Disciplinary Research*. Elsevier, 2022. pp. 159–185.
16. Bittencourt II, Costa E, Silva AP et al. A computational model for developing semantic web-based educational applications. *Educational Technology & Society* 2009; 12(1): 1–12.
17. Garcia-Gonzalez H, Gayo J and Paule-Ruiz M. Enhancing e-Learning Content by Using Semantic Web Technologies. *IEEE Transactions on Learning Technologies* 2017; 10(4): 554–550.
18. Dang D, Nguyen H, Ngo H et al. Information retrieval from legal documents with ontology and graph embeddings approach. In *Proceedings of 36th International conference on industrial, engineering and other applications of applied intelligent systems (IEA/AIE 2023)*. Springer, pp. 300–312.
19. Abu-Salih B and Alotaibi S. A systematic literature review of knowledge graph construction and application in education. *Heliyon* 2024; 10(3): e25383.
20. Nguyen H, Truong D, Vu S et al. Knowledge management for information querying system in education via the combination of rela-ops model and knowledge graph. *Journal of Cases on Information Technology* 2023; 25(1): 1–17. DOI:10.4018/JCIT.324113.
21. Pham V, Cao T, Pham XH et al. Towards Intelligent Legal Consultancy based on Integrating of Ontology and Knowledge Graph and Compare with LLMs. In *Proceedings of 13th Conference on Information Technology and Applications (CITA 2024)*, volume 2. pp. 134–147.
22. Peng C, Xia F, Naseriparsa M et al. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review* 2023; 56(11): 13071–13102.
23. Chikwendu IA et al. A comprehensive survey on deep graph representation learning methods. *Journal of Artificial Intelligence Research* 2023; 78: 287–356.

24. Su Y, Cheng Z, Wu J et al. Graph-based cognitive diagnosis for Intelligent Tutoring Systems. *Knowledge-Based Systems* 2022; 253: 109547. DOI:10.1016/j.knosys.2022.109547.
25. Ma X, Liu Q, Ma S et al. Empowering adaptive MOOCs: A learning path recommendation model based on multi-level knowledge graphs and rule-guided path constraint optimization. *Information Processing & Management* 2027; 64(2): 105198. DOI:10.1016/j.ipm.2026.105198.
26. Xia X and Qi W. Multilayer knowledge graph construction and learning behavior routing guidance based on implicit relationships of moocs. *Technological Forecasting and Social Change* 2024; 204: 123442.
27. Chen H. Research and Design of Knowledge Graph-based Online Learning Platform for English Majors. In *Proc. 2nd International Conference on Education Digitalization and Computer Science (EDCS 2025), Guangdong-Hong Kong-Macao*. pp. 100–107.
28. Truong DM, Nguyen HD, Vu S et al. Construct an intelligent querying system in education based on ontology integration. In *2022 IEEE International Conference on Computing (ICOCO 2022)*. IEEE, pp. 340–345.
29. Truong D, Nguyen H, Tran NP et al. Intelligent problem solver in database systems based on ontology integration through text-to-sql. *Fusion: Practice and Application* 2024; 15(2): 121–131.
30. Hohenecker P and Lukasiewicz T. Ontology reasoning with deep neural networks. *Journal of Artificial Intelligence Research* 2020; 68: 503–540.
31. Delgrande JP et al. Current and future challenges in knowledge representation and reasoning. *Dagstuhl Reports* 2023; 12(7): 62–79.
32. Cheng K, Ahmed NK, Rossi RA et al. Neural-symbolic methods for knowledge graph reasoning: A survey. *ACM Transactions on Knowledge Discovery from Data* 2025; 18(9): Article 25.
33. Schlichtkrull M, Kipf TN, Bloem P et al. Modeling Relational Data with Graph Convolutional Networks. In *Proceedings of the 15th International Semantic Web Conference (ESWC 2018), Heraklion, Crete, Greece*. pp. 593–607.
34. Pan Y, Liu J, Zhang L et al. Inductive relation prediction with logical reasoning using contrastive representations. In *Proceedings of the 2022 conference on Empirical methods in Natural Language Processing (EMNLP 2022)*. pp. 4261–4274.
35. Han Y and Wang Z. Re-gnn: A new model for predicting circuit reliability degradation. In *International Conference of Pioneering Computer Scientists, Engineers and Educators*. pp. 269–280.
36. Ma J, Liu B, Li K et al. A review of graph neural networks and pretrained language models for knowledge graph reasoning. *Neurocomputing* 2024; : 128490.
37. Trouillon T, Welbl J, Riedel S et al. Complex embeddings for simple link prediction. In *Proceedings of 33rd International Conference on Machine Learning (ICML 2016)*. pp. 2071–2080.
38. Bordes A, Usunier N, Garcia-Duran A et al. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. pp. 2787–2795.
39. Sun Z, Deng ZH, Nie JY et al. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.
40. Joshi A, Kale S, Chandel S et al. Likert scale: Explored and explained. *British journal of applied science & technology* 2015; 7(4): 396.