International Journal of Machine Learning and Cybernetics (2025) 16:1971–1990
https://doi.org/10.1007/s13042-024-02372-4

ORIGINAL ARTICLE

Integrating global semantics and enhanced local subgraph for inductive link prediction

Xinyu Liang[1] · Guannan Si[1] · Jianxin Li[1] · Zhaoliang An[1] · Pengxin Tian[1] · Fengyu Zhou[2] · Xiaoliang Wang[3]

Received: 22 August 2023 / Accepted: 26 August 2024 / Published online: 6 September 2024

Abstract

Inductive link prediction (ILP) predicts missing triplets involving unseen entities in knowledge graphs (KGs). Existing ILP research mainly addresses seen-unseen entities in the original KG (semi-inductive link prediction) and unseen-unseen entities in emerging KGs (fully-inductive link prediction). Bridging-inductive link prediction, which focuses on unseen entities that carry evolutionary information from the original KG to the emerging KG, has not been extensively studied so far. This study introduces a novel model called GSELI (integrating global semantics and enhanced local subgraph for inductive link prediction), which comprises three components. (1) The contrastive learning-based global semantic features (CLSF) module extracts relation-specific semantic features between the original and emerging KGs and employs semantic-aware contrastive learning to optimize these features. (2) The GNN-based enhanced local subgraph (GELS) module employs personalized PageRank (PPR)-based local clustering to sample tightly-related subgraphs and incorporates complete neighboring relations to enhance the topological information of subgraphs. (3) Joint contrastive learning and supervised learning training. Experimental results on various benchmark datasets demonstrate that GSELI outperforms the baseline models in both fully-inductive and bridging-inductive link predictions.

Keywords Inductive link prediction · Fully-inductive · Bridging-inductive · Unseen entities

1 Introduction

Knowledge graphs (KGs) are structured collections of factual triplets (entity, relation, entity), where each entity is depicted as a node, and each relation is denoted as an edge. KGs play a crucial role in knowledge-intensive tasks such as intelligent question answering [1], recommendation systems [2], and semantic search [3]. However, due to the exponential growth of real-world data, most existing KGs suffer from noise and incompleteness. For instance, in the open KG Freebase, approximately 71% of people lack birthplace information, and 99% do not have ethnic information [4]. Consequently, predicting missing facts within KGs, a task commonly referred to as link prediction (LP), has garnered substantial research interest.

Despite this success, LP remains challenging in various real-world scenarios. Traditional transductive LP assumes that all entities and relations are present during training at test time. As real-world KGs are dynamic, the emergence of unseen entities necessitates the retraining of the KGs. For instance, DBpedia observed a daily average addition of 200 entities between late 2015 and early 2016 [5].

✉ Guannan Si
sign@sdjtu.edu.cn

Xinyu Liang
liangxinyu0514@163.com

Jianxin Li
lijx7617@163.com

Zhaoliang An
anzhaoliang1@163.com

Pengxin Tian
tianpengxin1123@163.com

Fengyu Zhou
zhoufengyu@sdu.edu.cn

Xiaoliang Wang
xliang0505@163.com

[1] School of information Science and Electrical Engineering, Shangdong Jiaotong University, Jinan 250357, State, China

[2] School of Control Science and Engineering, Shandong University, Jinan 250000, State, China

[3] Shandong Longxihanzhang Technology Development Co., Ltd, Jinan 250013, State, China

However, the frequent increase of entities can result in a notable rise in operational overhead. In response to this challenge, certain scholars focus on inductive link prediction (ILP), aiming to predict missing triplets involving unseen entities without training KGs from scratch.

Recent years have seen the emergence of successful graph neural networks (GNNs), such as GraphSAGE [6] and GraphSAINT [7], for inductive embedding within graphs. GNN-based models [8–10] facilitate the embedding of unseen entities into the original KG by aggregating features of neighboring nodes. Logic rule-based methods inherently exhibit inductive capabilities for triplets involving unseen-unseen entities, owing to the entity-independent nature of logic rules. Subgraphs represent effective path combinations connecting two target entities, providing a more comprehensive and informative representation than a single rule. Subgraph-based approaches, which leverage subgraph topological information and exploit GNNs' embedding capacity, have gained substantial attention [11–16].

Ali et al. categorize ILP into two settings: semi-inductive and fully-inductive [17]. In a semi-inductive setting, out-of-knowledge KG (OOKG) entities, as shown in Fig. 1a, are associated with seen entities in the KG. As shown by the green dashed lines in Fig. 1, the auxiliary triplet (Lily, friend_of, Justin) and existing triplets are used to predict the triplet (Justin, aunt_of, Grace). In the fully-inductive setting, predicting triplets in the emerging KG involves unseen entities. As shown by the orange dashed lines in Fig. 1c, all entities in the triplet (Zeke, mother_of, Alisa) are unseen. The overlooked bridging inductive links connect seen entities in the original KG with unseen entities in the emerging KG, such as the predicted triplet (Chioe, friend_of, Zeke) shown by the red dashed lines in Fig. 1. However,"Chioe" and"Zeke"exist within disconnected KGs, lacking the connected subgraph surrounding the target entity. As a result, the aforementioned models [11–15] heavily depend on target entity connectivity to reason within the connected subgraph, rendering them incapable of handling bridging-inductive links. Furthermore, in 2015, law enforcement achieved a breakthrough by connecting a criminal case with another apparently unrelated case, showcasing the significance and feasibility of bridging-inductive links.

The latest research on DEKG-ILP [18] has extended subgraph-based approaches to achieve bridging-inductive links. Firstly, contrastive learning-based relation-specific feature modeling (CLRM) leverages global relational semantic information shared between the original KG and the emerging KG. Secondly, GNN-based subgraph modeling (GSM) utilizes local topological information within the KGs. However, GSM's focus on simple local subgraphs entails certain limitations: (1) During subgraph extraction, as the k-hop value increases, the number of sampled neighbors exponentially escalates, possibly weakening the expressive capacity of GNNs; (2) Solely considering local subgraphs may lead to incomplete neighboring relations. For instance, an enclosing subgraph consists of a subset of nodes and all their relations in a KG. As shown in Fig. 1b, the enclosing subgraph includes the entities"Grace","Chioe", and"David"along with the orange lines connecting them. Although the neighboring relations "aunt_of"and"friend_of"between"Grace"and"Chi oe"contain valuable information, they are not included in the enclosing subgraph. Additionally, when enclosing subgraphs are empty or sparse, GSELI's effectiveness may be compromised. For instance, in Fig. 1c, no viable paths connect the entities"Zeke" and"Alisa."

Based on the aforementioned observations, we introduce GSELI, a model that effectively integrates global semantics and enhanced local subgraphs to facilitate bridging-inductive links. Inspired by LCILP [15] and SNRI [14], GSELI enhances the GSM module within DEKG-ILP [18] and introduces the pioneering GNN-based enhanced local subgraph (GELS) module. Specifically, the GELS module employs local clustering based on personalized PageRank (PPR) to sample tightly related subgraphs and utilizes neighboring relational features and relational paths to enhance the topological information of the local subgraph. In summary, our main contributions can be summarized as follows:

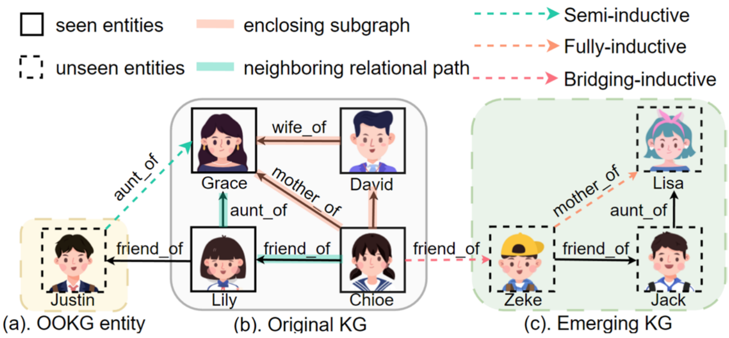

Fig. 1 Semi-inductive LP links OOKG entities to the original KG. Fully-inductive link prediction (LP) predicts missing links involving unseen-unseen entities in the emerging KG. The more challenging bridging-inductive LP predicts missing links involving unseen entities from the original KG to the emerging KG

- We are among the first scholars to study bridging-inductive link prediction and also conduct research on fully-inductive link prediction.
- A unified ILP framework, GSELI, integrates global semantics with enhanced local subgraphs, effectively addressing inductive link prediction tasks.
- We innovatively propose a GELS module, which uses PPR-based local clustering to sample tightly related subgraphs and integrates complete neighboring relations into the subgraphs.
- Experimental results across multiple benchmark datasets showcase GSELI's superiority over existing models in both fully-inductive and bridging-inductive link predictions.

The structure of the remaining sections of this paper is as follows: Sect. 2 reviews related work on link prediction and contrastive learning. Section 3 defines the terms used in this paper and classifies current inductive scenarios. Section 4 describes our proposed GSELI model in detail. Section 5 presents the experimental setup and results analysis. Section 6 discusses further experiments and analysis. Finally, Sect. 7 summarizes the findings.

2 Related work

This section describes related work, encompassing methods for both link prediction (see Sect. 2.1) and contrastive learning (see Sect. 2.2).

2.1 Link prediction methods

2.1.1 Transductive methods

Transductive LP requires that entities appearing during testing must have been seen during training. Translation-based methods [19, 20] treat relations as translational transformations projecting entities into lower-dimensional spaces. Tensor decomposition-based methods [21, 22] decompose KGs into three-way tensors for efficient computation of embedding representations. However, these methods overlook the hidden structural information present in KGs. In contrast, GNN-based approaches [23, 24] utilize GNNs to aggregate neighbor information for capturing global structural patterns. Unfortunately, the aforementioned methods necessitate retraining KGs from scratch when encountering unseen entities, leading to significant time and computational overhead.

2.1.2 Rule-based inductive methods

Rule-based methods offer inherent inductive and interpretable capabilities when predicting triplets involving unseen-unseen entities in emerging KGs, as logical rules are independent of entities. Traditional rule-based methods primarily focus on identifying good rules within KGs [25–27]. However, these methods face challenges in scalability and computational complexity. Although neural networks are known for their robustness and efficiency in addressing such issues [28–30], they struggle to scale to large KGs and fully leverage complex structural information, limiting their reasoning performance.

2.1.3 GNN-based inductive methods

Early researchers employed GNNs to aggregate seen neighboring entities around unseen entities to generate embeddings for the unseen entities [8–10]. However, these methods require unseen entities to be connected to seen entities, which does not address predicting missing triplets involving unseen-unseen entities in emerging KG. Recently, subgraph-based methods have rapidly advanced, leveraging subgraph topology information and the expressive power of GNNs [11–15]. However, the aforementioned models only predict links involving unseen entities in the original KG and the emerging KG separately. DEKG-ILP [18] further introduces bridging-inductive links that account for unseen entities carrying evolutionary information between the source KG and the emerging KG. However, this research is in its early stages and requires further exploration.

Table 1 summarizes the scenarios addressed by various LP models. Semi-inductive LP links a seen entity from the original KG to an unseen entity related to the original KG. Fully-inductive LP handles links between unseen-unseen entities, which are unrelated to the original KG. Bridging-inductive LP connects a seen entity from the original KG to an unseen entity unrelated to the original KG. In summary, semi-inductive LP leverages partial known information for higher accuracy and reliability, fully-inductive LP offers strong generalization and broad applicability, and bridging-inductive LP integrates known and unknown information, enhancing performance and adaptability in complex scenarios. The challenge levels are: bridging-inductive > fully-inductive > semi-inductive. Notably, DEKG-ILP and our proposed GSELI model handle both traditional LP tasks and more challenging bridging-inductive scenarios.

Table 1 Summary of applicable scenarios for link prediction models.The ✓ means you can handle the scenarios, and the ✗ means you can't

	Models	Transductive link prediction	Inductive link prediction		
			Semi-inductive	Fully-inductive	Bridging-inductive
Transductive models	TransE [19]	✓	✗	✗	✗
	RESCAL [21]	✓	✗	✗	✗
	RGCN [23]	✓	✗	✗	✗
Inductive models	MEAN [8]	✓	✓	✗	✗
	RuleN [26]	✓	✓	✓	✗
	GraIL [11]	✓	✓	✓	✗
	DEKG-ILP [18]	✓	✓	✓	✓
	GSELI (ours)	✓	✓	✓	✓

2.2 Contrastive learning

Contrastive learning (CL) is a self-supervised learning framework widely employed in natural language processing and computer vision. It aims to obtain representations by encoding similar or dissimilar samples. Recently, researchers have explored the application of CL in the field of ILP. SimKGC [31] introduces negative sampling (in-batch negatives, pre-batch negatives, and self-negatives) to increase the size of negative samples to thousands. RPC-IR [32] addresses the issue of deficient rule supervision in subgraphs by constructing positive and negative relational paths. SGI [33] maximizes the mutual information (MI) between the target relation and its enclosing subgraph, and it selects negative samples using a pre-trained MI estimator. SNRI [14] models subgraphs in a global manner by maximizing the subgraph-graph MI.

3 Definition

This section defines the original KG and the emerging KG (see Sect. 3.1) and categorizes LP tasks into transductive and inductive LP (see Sect. 3.2). Inductive link prediction is further subdivided into semi-inductive, fully-inductive, and bridging-inductive LP (see Sect. 3.3).

3.1 Original and emerging KG

Definition 1 (*Original knowledge graph*) The original KG is defined as: $\mathcal{G}(\mathcal{E},\mathcal{R}) = \{(h,r,t) \mid h,t \in \mathcal{E},\ r \in \mathcal{R}\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ represent the sets of entities and relations, respectively. The fact is in a triplet format (h, r, t), where h, t, and r represent the head entity, the tail entity, and the relation between them, respectively.

Definition 2 (*Emerging knowledge graph*) The emerging KG is defined as: $\mathcal{G}'(\mathcal{E}',\mathcal{R}) = \left\{(h,r,t) \mid h,t \in \mathcal{E}',\ r \in \mathcal{R}\right\} \subseteq \mathcal{E}' \times \mathcal{R} \times \mathcal{E}'$, where $\mathcal{E}' \cap \mathcal{E} = \emptyset$. The emerging KG $\mathcal{G}'(\mathcal{E}',\mathcal{R})$ consists of the unseen entity set $\mathcal{E}'$ and the seen relation set $\mathcal{R}$ shared with the original KG $\mathcal{G}(\mathcal{E},\mathcal{R})$.

3.2 Transductive and inductive LP

LP can be categorized into two scenarios: transductive and inductive LP.

Definition 3 (*Transductive link prediction*) Transductive LP is defined as predicting missing triplets $(h,r,t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where both the entity set $\mathcal{E}$ and the relation set $\mathcal{R}$ are seen.

Definition 4 (*Inductive link prediction*) Inductive LP is defined as predicting missing triplets $(h,r,t) \in (\mathcal{E} \cup \mathcal{E}^*) \times (\mathcal{R} \cup \mathcal{R}^*) \times (\mathcal{E} \cup \mathcal{E}^*)$ where (h, r, t) contains at least one unseen element (entity or relation). Here, $\mathcal{E} \cap \mathcal{E}^* = \emptyset$ and $\mathcal{R} \cap \mathcal{R}^* = \emptyset$, with $\mathcal{E}^*$ and $\mathcal{R}^*$ representing the sets of unseen entities and unseen relations, respectively.

3.3 Semi-inductive, fully-inductive, and bridging-inductive LP

Current research classifies inductive LP into three categories: semi-inductive (SI), fully-inductive (FI), and bridging-inductive (BI). This study focuses on scenarios where entities are unseen but relations are seen. Below are their definitions.

Definition 5 (*Semi-inductive link prediction*) Semi-inductive LP is defined as predicting the missing triplets $(h,r,t) \in (\mathcal{E} \times \mathcal{R} \times \tilde{\mathcal{E}}) \cup (\tilde{\mathcal{E}} \times \mathcal{R} \times \mathcal{E})$. Here, $\tilde{\mathcal{E}}$ represents the unseen entities, where $\mathcal{E} \cap \tilde{\mathcal{E}} = \emptyset$. In this scenario, either h or *t* is from the original KG, and the other is an unseen entity associated with the original KG (i.e., OOKG entity). As shown in Fig. 1, the triplet (unseen head entity"Justin", seen relation"aunt_of", unseen tail entity"Grace") is predicted.

Definition 6 (*Fully-inductive link prediction*) Fully-inductive LP is defined as predicting the missing triplets of the emerging KG $(h, r, t) \in \mathcal{E}' \times \mathcal{R} \times \mathcal{E}'$. In this scenario, both *h* and *t* are unseen entities that are not associated with the original KG, i.e., from the emerging KG. As shown in Fig. 1, the triplet (unseen head entity"Zeke", seen relation"mother_of", unseen tail entity "Alisa") is predicted.

Definition 7 (*Bridging-inductive link prediction*) Bridging-inductive LP is defined as predicting the missing triplets from the original KG to the emerging KG $(h, r, t) \in (\mathcal{E} \times \mathcal{R} \times \mathcal{E}') \cup (\mathcal{E}' \times \mathcal{R} \times \mathcal{E})$. In this scenario, either *h* or *t* is from the original KG, and the other is an unseen entity from the emerging KG that is not associated with the original KG. As shown in Fig. 1, the triplet (seen head entity"Chioe", seen relation"friend_of", unseen tail entity"Zeke") is predicted.

Notably, although both semi-inductive and bridging-inductive LP address the issue of seen-unseen entities, they differ significantly. In semi-inductive LP, the unseen entities are related to the original KG (i.e., OOKG entities). In contrast, in bridging-inductive LP, the unseen entities are not associated with the original KG (from an emerging KG), making bridging-inductive LP more challenging.

4 Methods

This section offers an overview of our proposed GSELI model(see Sect. 4.1), and provides a detailed introduction to its three constituent components: the CL-based global semantic features module (CLSF) (see Sect. 4.2), the GNN-based enhanced local subgraph module (GELS) (see Sect. 4.3), and the joint training strategy (see Sect. 4.4).

4.1 Overview of GSELI

The overview of our proposed model GSELI is shown in Fig. 2. The model is composed of three parts: CLSF, GELS,

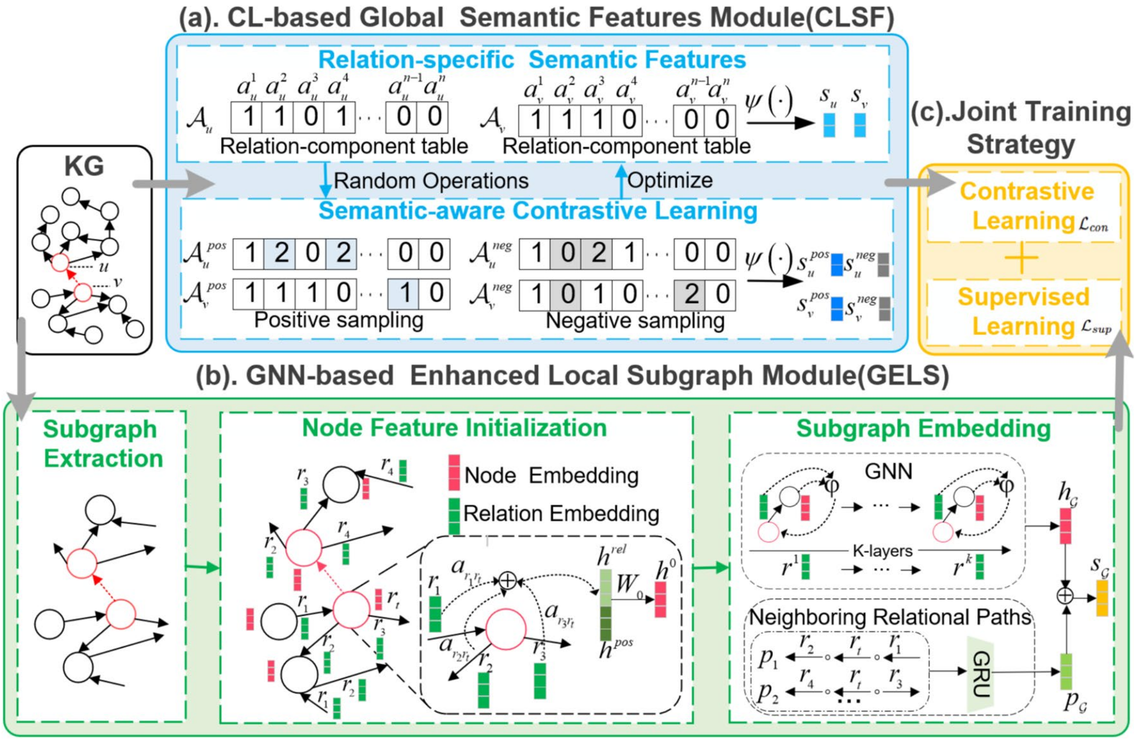
(a). CL-based Global Semantic Features Module(CLSF)
Relation-specific Semantic Features
$a_u^1\ a_u^2\ a_u^3\ a_u^4\ \ a_u^{n-1} a_u^n$
$\mathcal{A}_u$ 1 1 0 1 ··· 0 0
Relation-component table
$a_v^1\ a_v^2\ a_v^3\ a_v^4\ \ a_v^{n-1} a_v^n$
$\mathcal{A}_v$ 1 1 1 0 ··· 0 0
Relation-component table
$\psi(\cdot)$ s_u s_v
KG
u
v
Random Operations
Optimize
Semantic-aware Contrastive Learning
$\mathcal{A}_u^{pos}$ 1 2 0 2 ··· 0 0
$\mathcal{A}_v^{pos}$ 1 1 1 0 ··· 1 0
Positive sampling
$\mathcal{A}_u^{neg}$ 1 0 2 1 ··· 0 0
$\mathcal{A}_v^{neg}$ 1 0 1 0 ··· 2 0
Negative sampling
$\psi(\cdot)$ s_u^{pos} s_u^{neg} s_v^{pos} s_v^{neg}
(c).Joint Training Strategy
Contrastive Learning $\mathcal{L}_{con}$
+
Supervised Learning $\mathcal{L}_{sup}$
(b). GNN-based Enhanced Local Subgraph Module(GELS)
Subgraph Extraction
Node Feature Initialization
Node Embedding
Relation Embedding
r_1 r_2 r_3 r_4 r_t
$a_{r_1r_t}$ $a_{r_2r_t}$ $a_{r_3r_t}$
h^{rel} W_0 h^0 h^{pos}
Subgraph Embedding
GNN
φ
K-layers
r^1 r^k
$h_{\mathcal{G}}$
$s_{\mathcal{G}}$
Neighboring Relational Paths
p_1 p_2
GRU
$p_{\mathcal{G}}$

Fig. 2 Overview of GSELI. The CLSF module merges relation-component tables $\mathcal{A}_u$ and $\mathcal{A}_v$ to represent relation-specific semantic features $\boldsymbol{s}_u$ and $\boldsymbol{s}_v$. It further generates positive samples $\mathcal{A}_u^{pos}, \mathcal{A}_v^{pos}$ and negative samples $\mathcal{A}_u^{neg}, \mathcal{A}_v^{neg}$, forming $\boldsymbol{s}_u^{pos}, \boldsymbol{s}_v^{pos}$ and $\boldsymbol{s}_u^{neg}, \boldsymbol{s}_v^{neg}$, optimized through contrastive learning. The GELS module employs local clustering to identify the most relevant subgraphs, initializing node representations $\boldsymbol{h}_i^0$ with topological position $\boldsymbol{h}_i^{pos}$ and neighbor relation $\boldsymbol{h}_i^{rel}$ features. It combines local subgraph $\boldsymbol{h}_{\mathcal{G}}$ and neighboring relation path $\boldsymbol{p}_{\mathcal{G}}$ information to generate the subgraph representation $\boldsymbol{s}_{\mathcal{G}}$, and calculates the loss $\mathcal{L}$ through contrastive $\mathcal{L}_{con}$ and supervised $\mathcal{L}_{sup}$ learning

and a joint training strategy. Compared to existing studies [11–14, 16] that focus solely on fully-inductive LP, this model supports both fully-inductive and bridging-inductive LP.

Specifically, CLSF extracts globally shared relation-based semantic features across knowledge graphs and optimizes these features using a semantic-aware contrastive learning strategy. The CLSF module represents the relation-specific semantic features $\boldsymbol{s}_u$ and $\boldsymbol{s}_v$ by merging the relation component tables $\mathcal{A}_u$ and $\mathcal{A}_v$ of the target entities u and v. By generating positive samples $\mathcal{A}_u^{pos}$ and $\mathcal{A}_v^{pos}$, and negative samples $\mathcal{A}_u^{neg}$ and $\mathcal{A}_v^{neg}$ from these tables, it creates positive $\boldsymbol{s}_u^{pos}$ and $\boldsymbol{s}_v^{pos}$ and negative $\boldsymbol{s}_u^{neg}$ and $\boldsymbol{s}_v^{neg}$ relation-specific semantic features, which are further optimized through contrastive learning. Notably, by sharing the relational space, entities in both the original KG and emerging KGs are connected. This resolves the issue of lacking connected subgraphs around target entities, thereby improving bridging-inductive LP.

GELS extracts topological features from local subgraphs around target entities through three sub-modules: subgraph extraction, node feature initialization, and subgraph embedding. It first identifies the most relevant subgraphs using local clustering techniques, then initializes node representations $\boldsymbol{h}_i^0$ for node i with topological position features $\boldsymbol{h}_i^{pos}$ and neighbor relation features $\boldsymbol{h}_i^{rel}$ using a target relation-aware neighbor attention mechanism $\alpha r r_t$. Finally, it combines local subgraph information $\boldsymbol{h}_{\mathcal{G}}$ and neighboring relation path information $\boldsymbol{p}_{\mathcal{G}}$ to generate the final topological representation of the subgraph $\boldsymbol{s}_{\mathcal{G}}$. Notably, this module leverages local topological information around target entities in two disconnected KGs, facilitating both fully-inductive and bridging-inductive LP.

Finally, the model is trained using joint contrastive $\mathcal{L}_{con}$ and supervised learning $\mathcal{L}_{sup}$ losses, where the supervised learning computes the target link score by combining the semantic perspective learned from CLSF and the topological perspective learned from GELS.

4.2 CLSF module

4.2.1 Relation-specific semantic features

The semantics of entities in KGs are influenced by their associated relations. Following [18], we utilize relational features to represent the seen entities in the original KG. Given that both the original and the emerging KGs share relations $\mathcal{R}$, we can also employ these relational features to represent the unseen entities in the emerging KG. This approach allows us to embed both seen and unseen entities into a unified feature space, effectively resolving the issue of disconnected entities between the original and emerging KGs.

Specifically, we achieve the semantic representation $\boldsymbol{s}_i$ of entity i by integrating the relation-component table $\mathcal{A}_i$ of entity i with the relation-specific features $\mathcal{F}$ of the relation $\mathcal{R}$:

$$\boldsymbol{s}_i = \psi\left(\mathcal{A}_i, \mathcal{F}\right) = \frac{\sum_{t=1}^{n} a_i^t \cdot \boldsymbol{f}_t}{\sum_{t=1}^{n} a_i^t}, \tag{1}$$

$$\mathcal{A}_i = \left\{a_i^t \mid i \in \mathcal{E}, r_t \in \mathcal{R}\right\}, \tag{2}$$

$$\mathcal{F} = \left\{\boldsymbol{f}_t \mid r_t \in \mathcal{R}\right\}, \tag{3}$$

where $\psi(\cdot)$ is the fusion function, $|\mathcal{R}| = n$, $\boldsymbol{f}_t$ is the semantic embedding of relation r_t. $\mathcal{A}_i$ is a table that describes the semantic information of entity i by recording the number of triplets a_i^t associated with different relations r_t. If there are no triplets associated with relation r_t, a_i^t is set to 0. The relation-component tables $\mathcal{A}_u$ and $\mathcal{A}_v$ for entities u and v are shown in Fig. 2.

Notably, this method models data in an entity-independent manner, allowing for natural generalization to the representation of unseen entities. Finally, the semantic likelihood score $\varphi^{sem}(u, r_t, v)$ of the target triplet is calculated as follows:

$$\varphi^{sem}(u, r_t, v) = \left\langle \boldsymbol{s}_u, \boldsymbol{r}_t^{sem}, \boldsymbol{s}_v \right\rangle, \tag{4}$$

where $\boldsymbol{r}_t^{sem}$ is the learned embedding of relation r_t from the semantic perspective, and $\langle , , \rangle$ denotes the element-wise product of embedding vectors inspired by DistMult [22].

4.2.2 Semantic-aware contrastive learning

To optimize relation-specific features $\mathcal{F}$ during training, we utilize a contrastive learning-based approach along with a semantic-aware sampling strategy.

Following [18], we first model the semantic changes of entities by defining three random operations for the relation-component table $\mathcal{A}_i$ of each entity i: relation variation $o_1(\cdot)$, relation addition $o_2(\cdot)$, and relation deletion $o_3(\cdot)$. Specifically, operations $o_1(\cdot)$, $o_2(\cdot)$, and $o_3(\cdot)$ involve selecting a number from sets $\left\{a_i^t \mid a_i^t \in \mathcal{A}_i \wedge a_i^t \neq 0\right\}$, $\left\{a_i^t \mid a_i^t \in \mathcal{A}_i \wedge a_i^t = 0\right\}$, and $\left\{a_i^t \mid a_i^t \in \mathcal{A}_i \wedge a_i^t \neq 0\right\}$ respectively, and are then randomly assigned integers within the ranges $[1, m_i \cdot \theta]$, $[1, m_i \cdot \theta]$, and 0 respectively, where θ is the hyperparameter of the scaling factor, controlling the degree of random variation or addition of these relations, and m_i is the average number of triplets associated with each relation, expressed as:

$$m_i = \frac{\sum_{t=1}^{n} a_i^t}{\left|\left\{a_i^t \mid a_i^t \in \mathcal{A}_i \wedge a_i^t = 0\right\}\right|}. \tag{5}$$

Secondly, by employing these three random operations, we utilize operation $o_1(\cdot)$ to generate $\mathcal{A}_i^{pos}$, and apply both operations $o_2(\cdot)$ and $o_3(\cdot)$ to derive $\mathcal{A}_i^{neg}$. We subsequently produce the corresponding $\boldsymbol{s}_i^{pos}$ and $\boldsymbol{s}_i^{neg}$, denoted as:

$$\boldsymbol{s}_i^{pos} = \psi\left(\mathcal{A}_i^{pos}, \mathcal{F}\right), \quad \boldsymbol{s}_i^{neg} = \psi\left(\mathcal{A}_i^{neg}, \mathcal{F}\right). \tag{6}$$

Overall, the functions of the relation-component table are: (1) describing the semantic information of entities by recording their associations with various relations; (2) generating positive and negative samples during model training and optimizing the model through contrastive learning; and (3) constructing the semantic representation of entities by combining the relation-component table with relation-specific features. This approach enables the model to share relation features across different KGs, thereby enhancing its generalization ability.

Then the contrastive learning loss $\mathcal{L}_{con}$ is calculated to optimize the relation-specific features by maximizing the similarity between positive samples $\left(\boldsymbol{s}_i^{pos}, \boldsymbol{s}_i\right)$ and minimizing the similarity between negative samples $\left(\boldsymbol{s}_i^{neg}, \boldsymbol{s}_i\right)$, denoted as:

$$\mathcal{L}_{con} = \left[sim\left(\boldsymbol{s}_i^{pos}, \boldsymbol{s}_i\right) - sim\left(\boldsymbol{s}_i^{neg}, \boldsymbol{s}_i\right) + \gamma\right]_+, \tag{7}$$

where $sim(\cdot)$ represents a similarity function that measures the similarity between two embedded vectors by computing their Euclidean distance, γ is the hyperparameter controlling the margin, and $[x]_+ = \max\{0, x\}$.

4.3 GELS module

Algorithm 1 PPR-based subgraph extraction

```
Require: Adjacency matrix $A$, Seed nodes $\Omega = \{u, v\}$, Teleportation probability $\alpha$, Residual error $\epsilon$
Ensure: Local cluster subgraph $G(u, r_t, v)$
Initialize:
    PageRank vector $\mathbf{p} \leftarrow \mathbf{0}$
    Residual vector $\mathbf{r}[i] \leftarrow \frac{1}{|\Omega|}$ for $i \in \Omega$
    Queue $\leftarrow$ deque($\Omega$)
    Degree vector $\mathbf{d} \leftarrow \sum_{j=0}^{n-1} A_{ij}$
    Node volume $v \leftarrow 0$; $V \leftarrow \sum_{i=1}^{n} \mathbf{d}[i]$
    Cut value $c \leftarrow 0$
    Best conductance $\phi_{\text{best}} \leftarrow 1$
    Best community set $S_{\text{best}} \leftarrow \{\Omega[0]\}$
while queue is not empty do
    $i \leftarrow$ queue.pop()
    $\delta_{\text{pro}} \leftarrow \mathbf{r}[i] - 0.5\epsilon\mathbf{d}[i]$
    $\mathbf{r}[i] \leftarrow 0.5\epsilon\mathbf{d}[i]$
    $\mathbf{p}[i] \leftarrow \mathbf{p}[i] + (1-\alpha)\delta_{\text{pro}}$
    $\delta_{\text{dis}} \leftarrow \alpha\delta_{\text{pro}}$
    for each neighbor $k$ of $i$ do
        $\mathbf{r}_{\text{old}}[k] \leftarrow \mathbf{r}[k]$
        $\mathbf{r}[k] \leftarrow \mathbf{r}[k] + \delta_{\text{dis}} \frac{A_{ik}}{\mathbf{d}[i]}$
        if $\mathbf{r}[k] \geq \epsilon\mathbf{d}[k] \wedge \mathbf{r}_{\text{old}}[k] < \epsilon\mathbf{d}[k]$ then
            queue.append($k$)
        end if
    end for
end while
Sort vertices by score: $\mathbf{p}[i_1] \geq \mathbf{p}[i_2] \geq \ldots \geq \mathbf{p}[i_J]$
for node $i_j$ in sorted nodes do
    $v \leftarrow v + \mathbf{d}[i_j]$
    for neighbor $x$ in neighbors of $i_j$ do
        $c \leftarrow c - 1$ if $x \in S$ else $c + 1$
    end for
    $S \leftarrow S \cup \{i_j\}$
    if $v = V$ then
        break
    end if
    $\phi \leftarrow \frac{c}{\min(v, V-v)}$
    if $\phi < \phi_{\text{best}}$ then
        $\phi_{\text{best}} \leftarrow \phi$
        $S_{\text{best}} \leftarrow S$
    end if
end for
Prune nodes not on the path between $u$ and $v$, resulting in the final local cluster subgraph $\mathcal{G}(u, r_t, v)$.
```

4.3.1 Subgraph extraction

The PPR-based subgraph extraction algorithm involves two main steps. First, it employs an approximate PPR method to score nodes based on their proximity to a seed set of target entities. Second, it creates nested local clusters in descending order of these scores and evaluates them using conductance. The detailed procedure is provided in Algorithm 1.

The PageRank vector $\mathbf{p}$ is initialized to zero, and for each node i in the seed set Ω, the residual vector $\mathbf{r}[i]$ is set to $\frac{1}{|\Omega|}$. An empty double-ended queue (deque) is created to store nodes for processing. The degree vector $\mathbf{d}$ represents each node's degree and is obtained by summing the elements of the i-th column of the adjacency matrix A. Each element A_{ij} indicates the connection between node i and node j, $\mathbf{d}[i]$ denotes the degree of node i. The node volume v to 0 and V to the sum of the degrees of all nodes. The cut value c are set to zero. The initial best conductance ϕ_{best} is set to 1, and the best community set S_{best} is initialized with the first node in the seed set Ω. These steps lay the groundwork for the subsequent PPR-based subgraph extraction.

First, initialize a deque 'queue' to store nodes to be processed. In the main loop, when 'queue' is not empty, pop a node i from the deque and calculate δ_{pro}, the value that the current node will propagate to its neighboring nodes. Update the residual value $\mathbf{r}[i]$ of node i and its PageRank vector $\mathbf{p}[i]$. Next, compute δ_{dis} to be distributed to the neighboring nodes by multiplying δ_{pro} by the teleportation probability α. For each neighboring node k of node i, save its old residual value $\mathbf{r}_{\text{old}}[k]$ and update its residual value $\mathbf{r}[k]$. Calculate the threshold $(\epsilon \cdot \mathbf{d}[k])$ for neighboring node k. If the updated residual value exceeds the threshold and the old residual value is less than the threshold, add the neighboring node to the deque 'queue'. The main loop ends when the deque is empty. This process iteratively traverses and updates the PageRank values and residuals of each node and its neighbors until all nodes are processed, resulting in a vector $\mathbf{p}$ containing the approximate PPR values for each node.

Then the best sweep set is determined using PageRank scores and conductance calculations, resulting in a local cluster subgraph. First, vertices are sorted in descending order by PageRank scores: $p[i_1] \geq p[i_2] \geq \ldots \geq p[i_j]$. Next, the sorted vertices i_j are iterated. For each vertex, its volume v is updated by adding the degree $\mathbf{d}[i_j]$ of the current vertex i_j. The neighbors x of each vertex i_j are then iterated. Based on whether a neighbor is in the sweep set S, the cut value c is adjusted: if $x \in S$, c is decreased by 1; otherwise, c is increased by 1. The current vertex i_j is then added to the sweep set S. If the volume v reaches the specified value V, the loop terminates. The conductance ϕ of the current subset is then calculated and compared with the best conductance ϕ_{best}. If the current conductance is smaller, the best conductance ϕ_{best} and the best sweep set S_{best} are updated. Finally, by pruning nodes not on the path between seed nodes u and v, a tightly connected and highly relevant local cluster subgraph $\mathcal{G}(u, r_t, v)$ is obtained.

4.3.2 Node initialization

Since GNN requires a node feature matrix as input [34], ILP cannot utilize node attributes. Following [14], we initialize the node features $\boldsymbol{h}_i^0$ of the node i by combining topological locational features $\boldsymbol{h}_i^{pos}$ and neighboring relational features $\boldsymbol{h}_i^{rel}$.

Specifically, we initially acquire topological locational features $\boldsymbol{h}_i^{pos}$ through a double radius vertex labeling scheme, denoted as:

$$\boldsymbol{h}_i^{pos} = [\text{one-hot}(d(i,u)) \oplus \text{one-hot}(d(i,v))], \tag{8}$$

where the two target nodes u and v are uniquely labeled as (0,1) and (1,0), respectively, $d(i, u)$ is the shortest distance from the node i to the target node u without any path through v (similarly, $d(i, v)$ is the shortest distance to the target node v without any path through u), and $\oplus$ denotes the concatenation of two vectors. To address the topological limitations between the original KGs and the emerging KGs, and facilitate the implementation of bridging-inductive links, following [18], we retain the nodes $\{i \mid d(i,u) > t \vee d(i,v) > t\}$ in t-top while setting $d(i,\cdot) = -1$ and $\text{one-hot}(-1) = 0$ if $d(i,\cdot) > t$.

Secondly, we employ a target relation-aware neighbor attention mechanism for message passing on the node i to acquire neighboring relational features $\boldsymbol{h}_i^{rel}$, expressed as:

$$\boldsymbol{h}_i^{rel} = \sum_{r \in \mathcal{N}_r(i)} \alpha_{rr_t} \boldsymbol{r}_t, \tag{9}$$

$$\alpha_{rr_t} = \text{softmax}(\boldsymbol{r}, \boldsymbol{r}_t) = \frac{\exp(\boldsymbol{r}^\top \boldsymbol{r}_t)}{\sum_{r' \in \mathcal{N}_r(i)} \exp(\boldsymbol{r'}^\top \boldsymbol{r}_t)}, \tag{10}$$

where $\mathcal{N}_r(i)$ is the set of direct neighbors of the node i under relation r, α_{rr_t} is the importance of relation r to the node i under target relation r_t, and $\boldsymbol{r}$ and $\boldsymbol{r}_t$ represent the embeddings of the neighbor relation r and the target relation r_t respectively.

Finally, the initial embedding of the node i is defined as:

$$\boldsymbol{h}_i^0 = \boldsymbol{W}_0[\boldsymbol{h}_i^{pos} \oplus \boldsymbol{h}_i^{rel}], \tag{11}$$

where $\boldsymbol{W}_0$ is the corresponding transformation matrix.

4.3.3 Subgraph embedding

To address the issue of sparse subgraphs, we merge the extracted local subgraph $\boldsymbol{h}_{\mathcal{G}(u,r_t,v)}$ with the neighboring relational paths $\boldsymbol{p}_{\mathcal{G}(u,r_t,v)}$ to form a comprehensive representation of the subgraph's topology information $\boldsymbol{s}_{\mathcal{G}(u,r_t,v)}$.

Firstly, we input the target triplet (u, r_t, v) of the subgraph $\mathcal{G}(u, r_t, v)$ into the GNN to encode the extracted local subgraph. Specifically, we account for the interaction between nodes and relations, and express the node aggregation function at GNN's kth layer as:

$$\boldsymbol{h}_i^k = \sum_{r\in\mathcal{R}} \sum_{j\in\mathcal{N}_r(i)} \alpha_{irj}^k \boldsymbol{W}_r^k \varphi(\boldsymbol{r}^{k-1}, \boldsymbol{h}_j^{k-1}), \tag{12}$$

$$\alpha_{irj}^k = \sigma_1(\boldsymbol{W}_1^k \boldsymbol{s}_{ir} + \boldsymbol{b}_1^k), \tag{13}$$

$$\boldsymbol{s}_{ir} = \sigma_2(\boldsymbol{W}_2[\boldsymbol{h}_i^{k-1} \oplus \boldsymbol{h}_j^{k-1} \oplus \boldsymbol{r}^{k-1} \oplus \boldsymbol{r}_t^{k-1}] + \boldsymbol{b}_2^k), \tag{14}$$

where $\mathcal{R}$ is the set of relations in the KG, α_{irj}^k denotes the attention weights of edges connecting nodes i and j via relation r at layer k, $\boldsymbol{W}_r^k$ is the transformation matrix of relation r for propagating messages, $\varphi(\boldsymbol{r}_t^{k-1}, \boldsymbol{h}_j^{k-1})$ is the fusion operation to share the hidden features of nodes and relations, and σ_1 and σ_2 are the activation functions. Simultaneously, upon updating the node embeddings as defined in Eq. (12), the relation embedding also undergoes a transformation to project both nodes and relations into a shared embedding space, denoted as:

$$\boldsymbol{r}^k = \boldsymbol{W}_{\text{rel}}^k \boldsymbol{r}^{k-1}. \tag{15}$$

Given K as the number of GNN layers, the representation of the entire extracted local subgraph $\boldsymbol{h}_{\mathcal{G}(u,r_t,v)}$ is obtained by averaging the node representations as follows:

$$\boldsymbol{h}_{\mathcal{G}(u,r_t,v)} = \frac{1}{|\mathcal{V}_{\mathcal{G}(u,r_t,v)}|} \sum_{i\in\mathcal{V}_{\mathcal{G}(u,r_t,v)}} \boldsymbol{h}_i^K, \tag{16}$$

where $\mathcal{V}_{\mathcal{G}(u,r_t,v)}$ represents the set of nodes in the subgraph.

Secondly, we extract relational sequences among target nodes using an attention mechanism and a Gated Recurrent Unit (GRU) [35] to model the representations of neighboring relational paths $\boldsymbol{p}_{\mathcal{G}(u,r_t,v)}$, as expressed below:

$$\boldsymbol{p}_{\mathcal{G}(u,r_t,v)} = \sum_{p\in\mathcal{P}_{r_t}(u,v)} a_p \boldsymbol{p}, \tag{17}$$

$$a_p = \text{softmax}(\boldsymbol{p}, \boldsymbol{r}_t) = \frac{\exp(\boldsymbol{p}^\top \boldsymbol{r}_t)}{\sum_{p'\in\mathcal{P}(u,v)} \exp(\boldsymbol{p}'^\top \boldsymbol{r}_t)}, \tag{18}$$

$$\boldsymbol{p} = \text{GRU}(p) = \text{GRU}(\boldsymbol{r}_u, \boldsymbol{r}_t, \boldsymbol{r}_v), \tag{19}$$

where $\mathcal{P}_{r_t}(u, v)$ denotes the set of all neighboring relational paths between target nodes u and v under the target relation r_t. The final representation of the subgraph's topological information $\boldsymbol{s}_{\mathcal{G}(u,r_t,v)}$ is as follows:

$$\boldsymbol{s}_{\mathcal{G}(u,r_t,v)} = \boldsymbol{h}_{\mathcal{G}(u,r_t,v)} \oplus \boldsymbol{p}_{\mathcal{G}(u,r_t,v)}. \tag{20}$$

Importantly, we integrate complete neighboring relations into the subgraph from two aspects: the neighboring relational features of node features and the neighboring relational paths of the sparse subgraph. Finally, the topological likelihood score $\varphi^{tpo}(u, r_t, v)$ of the target triplet (u, r_t, v) is given by:

$$\varphi^{tpo}(u, r_t, v) = \boldsymbol{W}_s[\boldsymbol{h}_u^K \oplus \boldsymbol{h}_v^K \oplus \boldsymbol{r}_t^{tpo} \oplus \boldsymbol{s}_{\mathcal{G}(u,r_t,v)}], \tag{21}$$

where $\boldsymbol{r}_t^{tpo}$ is the learned embedding of relation r_t from the topological perspective.

Algorithm 2 Training process of GSELI

```
Require: Knowledge graph $\mathcal{G}$ and target triple $(u, r_t, v)$.
Ensure: Prediction score for $(u, r_t, v)$.
Extract relation-specific semantic features $s_u$ and $s_v$ of entities $u$ and $v$ (Eqs. (1), (2), (3));
Compute semantic likelihood score $\varphi^{sem}(u, r_t, v)$ (Eq. (4));
Optimize $s_u$ and $s_v$ via semantic-aware contrastive loss $\mathcal{L}_{con}$ (Eqs. (5), (6), (7));
Extract PPR-based local clustering subgraph $\mathcal{G}(u, r_t, v)$ (Algorithm. (1));
Initialize node features $\boldsymbol{h}_i^0$ with local topological locational features $\boldsymbol{h}_i^{pos}$ and neighboring relational
    features $\boldsymbol{h}_i^{rel}$ features (Eqs. (8), (9), (10), (11));
Get embedding of local subgraph $\boldsymbol{h}_{\mathcal{G}(u,r_t,v)}$ using GNN (Eqs. (12), (13), (14), (15), (16));
Get embedding of neighboring relational paths $\boldsymbol{p}_{\mathcal{G}(u,r_t,v)}$ (Eqs. (17), (18), (19));
Combine $\boldsymbol{h}_{\mathcal{G}(u,r_t,v)}$ and $\boldsymbol{p}_{\mathcal{G}(u,r_t,v)}$ to complete subgraph topology embedding $\boldsymbol{s}_{\mathcal{G}(u,r_t,v)}$ (Eq. (20));
Compute topological likelihood score $\varphi^{tpo}(u, r_t, v)$ (Eq. (21));
Compute final scores $\varphi(u, r_t, v)$ combining $\varphi^{sem}(u, r_t, v)$ and $\varphi^{tpo}(u, r_t, v)$ (Eq. (23));
Calculate supervised loss $\mathcal{L}_{sup}$ (Eqs. (22), (24));
Minimize final loss $\mathcal{L}$ by combining $\mathcal{L}_{sup}$ and $\mathcal{L}_{con}$ (Eq. (25)).
```

4.4 Joint training strategy

Our work aims to achieve the ultimate goal through a combination of supervised loss $\mathcal{L}_{sup}$ and contrastive loss $\mathcal{L}_{con}$.

Firstly, for supervised learning, we treat all triplets in the original KG as positive triplets $\mathcal{T}_{pos}$ and construct negative triplets $\mathcal{T}_{neg}$ by randomly replacing head or tail entities with another entity in $\mathcal{E}$. Formally, the set of negative triplets $\mathcal{T}_{neg}$ is defined as:

$$\mathcal{T}_{neg} = \{(u', r_k, v) \mid u' \in \mathcal{E}\} \cup \{(u, r_k, v') \mid v' \in \mathcal{E}\}. \tag{22}$$

Secondly, the total score for the supervised learning $\varphi(u, r_t, v)$ of the target triplet (u, r_t, v) is calculated by combining the global semantic information from Eq. (4) and the local topological information from Eq. (21). Formally, it is defined as:

$$\varphi(u, r_t, v) = \varphi^{sem}(u, r_t, v) + \varphi^{tpo}(u, r_t, v). \tag{23}$$

Its margin-based loss function is defined as follows:

$$\mathcal{L}_{sup}(u, r_t, v) = [\varphi(u', r_t, v') - \varphi\big(u, r_t, v\big) + \gamma]_+. \tag{24}$$

The final loss is calculated by combining the supervised learning loss from Eq. (24) and the contrastive learning loss from Eq. (7), and it is minimized through joint training, expressed as:

$$\mathcal{L} = \sum_{\mathcal{T}_{pos}} \sum_{\mathcal{T}_{neg}} \mathcal{L}_{sup} + \lambda \sum_{\mathcal{T}_{pos}} \mathcal{L}_{con}, \tag{25}$$

Table 2 Statistics of the inductive datasets proposed by DEKG: $|\mathcal{R}|$, $|\mathcal{E}|$, and $|\mathcal{T}|$ represent the number of relations, entities, and triplets, respectively

	Train			test_mix			test_bridging			test_fully		
	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{T}\rvert$
FB15K-237												
EQ	180	1594	4734	180	2687	6648	180	2687	6443	142	1093	2198
MB	200	2608	10,905	200	4268	15,318	200	4268	14,840	172	1660	4623
MF	215	3668	20,180	215	6169	26,689	215	6169	25,824	183	2501	8271
NELL-995												
EQ	14	3088	5101	14	3312	5718	14	3312	5618	14	225	933
MB	88	2544	9141	88	4551	14,201	88	4551	13,727	79	2074	5062
MF	142	4539	18,240	142	7914	25,625	142	7914	24,820	122	3514	8857
WN18RR												
EQ	9	2746	6040	9	3668	7404	9	3668	7216	8	922	1806
MB	10	6954	17,100	10	9711	20,596	10	9711	20,155	10	2757	4452
MF	11	12,078	28,998	11	17,154	33,130	11	17,154	32,525	11	5084	6932

Table 3 Statistics of the fully-inductive datasets proposed by GraIL: $|\mathcal{R}|$, $|\mathcal{E}|$, and $|\mathcal{T}|$ represent the number of relations, entities, and triplets, respectively

	FB15K-237			WN18RR		
	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{T}\rvert$
v1						
Train	180	1594	5226	9	2764	6678
Test	142	1093	2404	8	922	1991
v2						
Train	200	2608	12,085	10	6954	18,968
Test	172	1660	5092	10	2757	4863
v3						
Train	215	3668	22,394	11	12,078	32,150
Test	183	2501	9137	11	5084	7470
v4						
Train	219	4707	33,916	9	3861	9842
Test	200	3501	14,554	9	7084	15,157

where λ represents a hyper-parameter adjusting the proportion between the supervised learning and contrastive learning losses.

5 Experiments

This section,“Experiments,”includes the following parts: dataset (see Sect. 5.1), experimental details (see Sect. 5.2), results on DEKG datasets (see Sect. 5.3), results on GraIL datasets (see Sect. 5.4), and ablation study (see Sect. 5.5).

5.1 Dataset

We adhere to the benchmark datasets proposed in DEKG-ILP for fully inductive and bridging inductive link prediction. DEKG-ILP extends the FB15K237 [36], NELL-995 [37], and WN18RR [38] datasets (v1, v2, and v3) introduced in Grail by creating EQ (equal inductive links), MB (more bridging-inductive links), and MF (more fully-inductive links) versions. For example, we use the FB15K237 v1, v2, and v3 training sets to construct the EQ, MB, and MF training sets and the FB15K237 v1, v2, and v3 test sets to construct the EQ, MB, and MF fully-inductive test sets. The bridging-inductive evaluation datasets are created using the training sets and fully-inductive test sets, where one entity is in the training set, and the other is in the fully-inductive test set. We also construct the EQ, MB, and MF mixed evaluation datasets using the fully-inductive and bridging-inductive evaluation datasets in ratios of 1:1, 1:2, and 2:1. The relevant statistical data are listed in Table 2. Additionally, we use the v1, v2, v3, and v4 datasets of FB15K237 and WN18RR from GraIL to comprehensively evaluate the fully-inductive LP performance. The statistical data are presented in Table 3.

5.2 Experimental details

5.2.1 Evaluation protocol

We follow the evaluation protocol used in DEKG-ILP [18], which includes mean reciprocal rank (MRR) and Hits@N metrics (where N=1, 5, 10). We conducted evaluations for both relation prediction (h, ?, t) and entity prediction (?, r, t) and (h, r, ?) tasks. MRR represents the average reciprocal rank of all test triplets, while Hits@N measures the proportion of correctly ranked entities and relations within the top N. Negative triplets are obtained by replacing elements in the triplets with candidate sets of entities and relations that include all entities and relations in the KG. It is worth noting that higher MRR and Hits@N values indicate better link prediction performance. We conducted five experiments using different random seeds and recorded the average results.

5.2.2 Hyper-parameter setting

Our experiment is implemented using PyTorch with the Adam optimizer [39]. For subgraph extraction, we use a 3-hop subgraph. The hyperparameters are manually specified during the training process: a learning rate of 0.01, an edge dropout rate of 0.5, a relation-specific embedding dimension of 32, and a margin γ of 10 in the loss function for Eqs. (7) and (24). Additionally, we select hyperparameters θ and λ, which correspond to adjusting the scaling factor of the number of relations in the relational component table in Sect. 4.2.2 and adjusting the loss coefficient of the weight for contrastive learning in Eq. (25), respectively. We choose values for θ and λ from the range [0.1, 0.9], and the rationale for this selection will be explained in Sect. 6.2.

Table 4 The results of MRR and Hits@N (N = 1, 5, 10) on the EQ, MB, and ME versions of the FB15K-237, NELL-995, and WN18RR datasets

Model	EQ				MB				MF				
	MRR	H@1	H@5	H@10	MRR	H@1	H@5	H@10	MRR	H@1	H@5	H@10	Avg
MRR and Hits@N (N = 1, 5, 10) results on the FB15K-237 datasets													
Grail	0.282	0.223	0.319	0.339	0.270	0.213	0.291	0.301	0.509	0.432	0.586	0.616	0.365
TATC	0.318	0.263	0.345	0.354	0.276	0.227	0.296	0.305	0.484	0.484	0.411	0.595	0.363
CoMPILE	0.316	0.256	0.353	0.375	0.282	0.230	0.301	0.314	0.522	0.452	0.589	0.608	0.383
SNRI	0.362	0.281	0.419	0.496	0.313	0.224	0.375	0.478	0.415	0.303	0.528	0.620	0.401
RMPI	0.327	0.253	0.378	0.441	0.288	0.226	0.320	0.359	0.517	0.441	0.579	0.640	0.397
DEKG-ILP	0.421	0.285	0.572	0.699	0.542	0.409	0.693	0.825	0.632	0.511	0.777	0.889	0.605
GSELI(Ours)	**0.502**	**0.350**	**0.682**	**0.834**	**0.616**	**0.473**	**0.790**	**0.903**	**0.666**	**0.542**	**0.814**	**0.934**	**0.676**
MRR and Hits@N (N = 1, 5, 10) results on the NELL-995 datasets													
Grail	0.463	0.400	0.520	0.520	0.306	0.220	0.381	0.429	0.595	0.516	0.677	0.700	0.477
TATC	0.503	**0.435**	0.552	0.560	0.348	0.227	0.401	0.440	0.593	0.504	0.690	0.718	0.498
CoMPILE	0.475	0.422	0.517	0.517	0.341	0.283	0.364	0.382	0.629	**0.558**	0.698	0.718	0.492
SNRI	**0.505**	0.402	0.582	0.702	0.309	0.218	0.371	0.458	0.551	0.456	0.655	0.718	0.435
RMPI	0.486	0.411	0.547	0.607	0.347	0.278	0.389	0.468	**0.656**	0.579	**0.734**	0.785	0.524
DEKG-ILP	0.449	0.320	0.576	0.740	0.387	0.226	0.593	0.727	0.519	0.397	0.662	0.737	0.528
GSELI(Ours)	0.468	0.333	**0.615**	**0.763**	**0.465**	**0.327**	**0.637**	**0.746**	0.454	0.295	0.651	**0.784**	**0.545**
MRR and Hits@N (N = 1, 5, 10) results on the WN18RR datasets													
Grail	0.423	0.392	0.420	0.420	0.293	0.267	0.272	0.272	0.412	0.373	0.430	0.431	0.367
TATC	0.423	0.401	0.420	0.420	0.285	0.269	0.272	0.272	0.418	0.388	0.429	0.430	0.369
CoMPILE	0.421	0.400	0.417	0.417	0.274	0.255	0.260	0.260	0.399	0.367	0.400	0.400	0.323
SNRI	0.462	0.405	0.489	0.545	0.356	0.294	0.379	0.455	0.444	0.383	0.474	0.556	0.437
RMPI	**0.474**	**0.422**	0.494	0.539	0.354	**0.312**	0.379	0.437	**0.468**	**0.429**	0.477	0.505	0.441
DEKG-ILP	0.405	0.289	0.544	0.690	0.351	0.224	0.493	0.627	0.381	0.258	0.503	**0.715**	0.457
GSELI(Ours)	0.418	0.293	**0.563**	**0.718**	**0.356**	0.228	**0.493**	**0.642**	0.368	0.243	**0.503**	0.703	**0.461**

The EQ, MB, and ME test sets include both fully inductive and bridging inductive links, distributed in ratios of 1:1, 1:2, and 2:1, respectively. The best results are in bold, and the second-best are underlined

5.2.3 Baselines

We evaluated our model using the baseline models: GraIL, TACT, CoMPILE, SNRI, RMPI, and DEKG-ILP. These models use subgraph methods for ILP and support fully-inductive LP. Additionally, DEKG-ILP is designed specifically for bridging-inductive LP. We used publicly available source code and implemented these models with optimal parameters.

Tailored for fully-inductive LP are as follows:

- GraIL [11] has become the first model used for fully-inductive links by employing local subgraph sampling, subgraph node labeling, and GNN scoring.
- TACT [12] extends GraIL by categorizing all pairs of relations into seven topological patterns, aiming to leverage the topology-aware correlations between relations.
- CoMPILE [13] extends GraIL by introducing a new node-edge communication mechanism to enhance relational information.
- SNRI [14] extends GraIL by integrating complete neighboring relations into enclosing subgraphs, maximizing local–global MI between subgraph-graph to globally model neighboring relations.
- RMPI [16] extends GraIL through effective relational message passing and relational patterns to achieve subgraph reasoning.

Tailored for bridging-inductive LP are as follows:

- DEKG-ILP [18] is the first model to leverage subgraphs and global relational semantic features shared between the original KG and emerging KG for bridging-inductive LP.

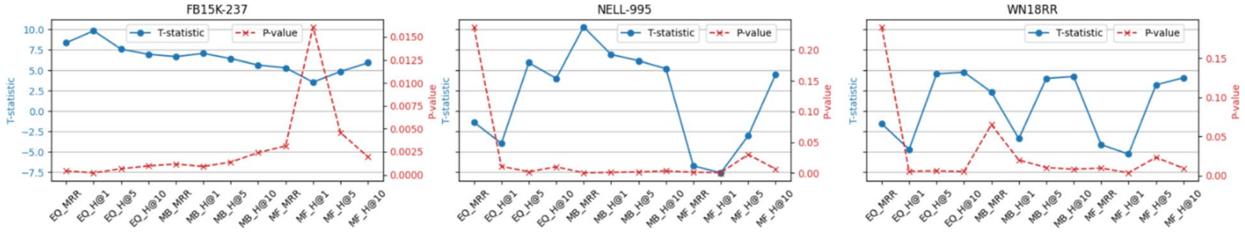

Fig. 3 Statistical significance analysis of MRR and Hits@N (N=1, 5, 10) on EQ, MB, and ME versions of FB15K-237, NELL-995, and WN18RR datasets

5.3 Results on DEKG datasets

5.3.1 Main results

Table 4 presents a comparison of the MRR and Hits@N (1, 5, 10) results of our proposed GSELI model with the baseline models on the FB15k-237, NELL-995, and WN18RR datasets using the EQ, MB, and MF versions. The test datasets include fully-inductive and bridging-inductive links in ratios of 1:1, 1:2, and 2:1. The results indicate that the GSELI model significantly outperforms the baseline models on most datasets. From Table 4, it can be observed that:

(1) GSELI shows more significant improvements on the FB15k-237 and NELL-995 datasets compared to WN18RR, particularly in the FB15k-237 dataset, where both MRR and Hits@N have increased. This improvement can be attributed to the larger number of relations in FB15k-237 and NELL-995, which allow GSELI to extract richer relation-specific semantic features and more comprehensive neighborhood relation information.

(2) Compared to the EQ and MF versions, GSELI and DEKG-ILP demonstrate more significant improvements in the MB version due to its inclusion of more bridging-inductive links. This improvement is attributed to the optimization of GSELI and DEKG-ILP for bridging-inductive links, while also maintaining strong performance in fully-inductive links. In contrast, other baseline models are specifically designed for fully-inductive links.

(3) GSELI shows limited ranking capabilities in the EQ and MF versions of the NELL-995 and WN18RR datasets. Its high-precision metrics (Hits@1 and Hits@5) are lower than those of fully-inductive LP models, although it surpasses other baseline models in the Hits@10 metric. This limitation may stem from the complexity and diversity of the EQ and MF datasets. These versions typically include more varied and intricate relational patterns, which GSELI fails to fully capture, leading to subpar high-precision performance. Additionally, GSELI is better suited for capturing broad relational patterns rather than the fine distinctions needed for high-precision ranking.

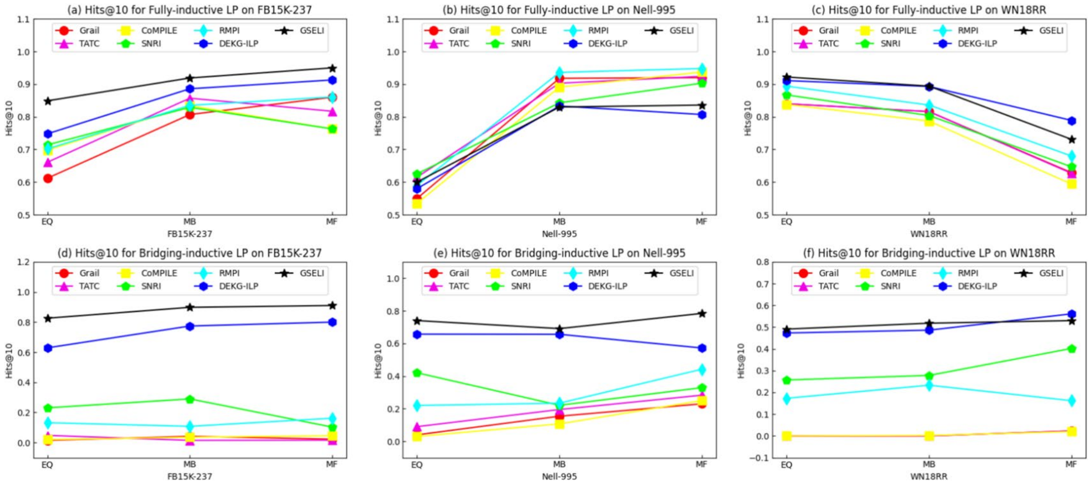

Fig. 4 Hits@10 results on the EQ, MB, and MF of the FB15k-237, NELL-995, and WN18RR datasets, including tests with solely fully-inductive links and solely bridging-inductive links

Figure 3 presents the statistical significance analysis of the EQ, MB, and MF versions of the FB15K-237, NELL-995, and WN18RR datasets on MRR and Hits@N (N = 1, 5, 10), comparing GSELI's performance against baseline models. A larger t-statistic (whether positive or negative) indicates a greater mean difference, while a smaller p-value suggests this difference is less likely due to random error. For the FB15K-237 dataset, all t-statistics are positive and significant, with p-values less than 0.005, indicating GSELI's superiority on all metrics. For the NELL-995 dataset, most EQ and MB metrics show significant differences, especially in MRR and Hits@1, with p values less than 0.001. However, the MF metrics perform worse on Hits@1 and Hits@5, with negative t-statistics, indicating GSELI's inferiority on these metrics. For the WN18RR dataset, the EQ and MB metrics are significant on Hits@1, Hits@5, and Hits@10, with p-values less than 0.01, but show no significant difference on MRR. Additionally, the MF metrics display significant negative differences, indicating that GSELI performs worse than the baseline models on some metrics in this dataset. Overall, GSELI generally outperforms the baseline models, though inconsistently.

5.3.2 Respective results

This section examines the Hits@10 results for the EQ, MB, and MF versions of the FB15k-237, NELL-995, and WN18RR datasets, focusing on both fully-inductive links and bridging-inductive links. Due to space limitations, we have not included the MRR, Hits@1, and Hits@5 metrics. According to Fig. 4, we observe the following:

(1) Overall, GSELI, leveraging contrastive learning-based global semantic features and GNN-based enhanced local subgraphs, outperforms most baseline models in Hits@10 across various benchmark datasets for both fully-inductive and bridging-inductive links, especially excelling in bridging-inductive link prediction.

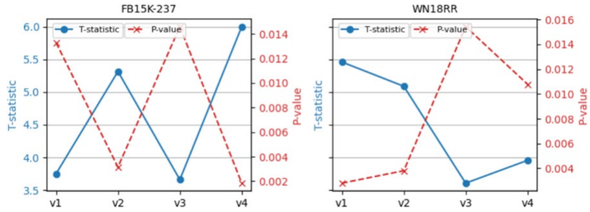

Fig. 5 Statistical significance analysis of Hits@10 on v1, v2, v3, and v4 versions of the FB15K-237, NELL-995, and WN18RR datasets

(2) In fully-inductive links, GSELI demonstrates excellent performance on the FB15k-237 and WN18RR datasets, thanks to its meticulous design. However, since GSELI is not specifically optimized for fully-inductive links, its performance on the NELL-995 dataset falls short compared to models explicitly designed for fully-inductive links, such as RMPI and GraIL.

(3) In bridging-inductive links, GraIL, TATC, and CoMPILE perform poorly due to topological limitations. SNRI demonstrates the ability to handle bridging-inductive links through global modeling of neighboring relations, while RMPI uses relational message-passing networks. GSELI and DEKG-ILP, by leveraging global semantic features, outperform SNRI and RMPI. However, GSELI surpasses DEKG-ILP by utilizing enhanced local subgraphs.

5.4 Results on GraIL datasets

To further evaluate fully-inductive link prediction, we used the FB15k-237 and WN18RR datasets in versions v1, v2, v3, and v4 proposed by GraIL and reported the Hits@10 results, as shown in Table 5.

Table 5 shows that our GSELI model performs exceptionally well across multiple versions of both datasets. On the FB15k-237 dataset, GSELI achieves the highest performance in versions v2 and v4 and the second-highest

Table 5 Hits@10 results on FB15k-237 and WN18RR for versions v1, v2, v3, and v4

Model	FB15k-237					WN18RR				
	v1	v2	v3	v4	Avg	v1	v2	v3	v4	Avg
Grail	64.15	81.80	82.83	89.29	79.52	84.25	78.68	58.43	73.41	73.69
TATC	62.20	80.02	84.16	88.41	78.70	82.45	78.68	58.60	73.41	73.29
CoMPILE	67.66	82.98	84.67	87.44	80.69	83.60	79.82	60.69	75.49	74.90
SNRI	71.79	86.50	89.59	89.39	84.32	87.23	83.10	67.31	83.32	80.24
RMPI	71.71	83.37	86.10	88.69	82.47	87.77	82.43	73.14	81.24	81.15
DEKG-ILP	**83.01**	<u>91.33</u>	**94.68**	<u>94.78</u>	<u>90.95</u>	<u>91.31</u>	<u>88.81</u>	<u>76.36</u>	<u>87.12</u>	<u>85.90</u>
GSELI(Ours)	<u>81.46</u>	**93.09**	<u>93.60</u>	**96.04**	**91.05**	**93.44**	**89.94**	**77.16**	**88.26**	**87.20**

The best results are in bold, and the second-best are underlined

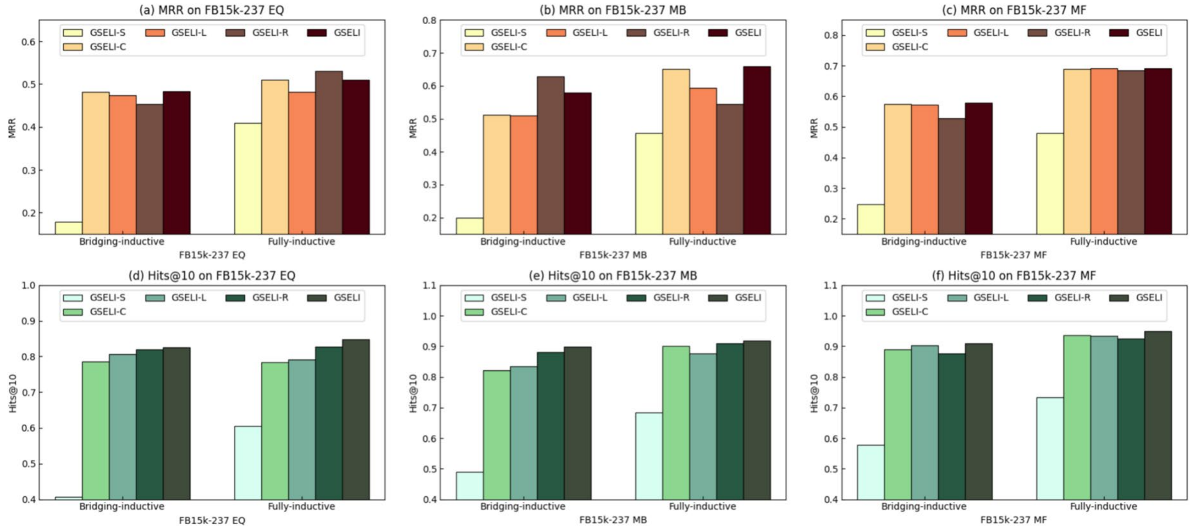

Fig. 6 MRR and Hits@10 results of ablation study on EQ, MB, MF of FB15k-237, for both solely fully-inductive links and solely bridging-inductive links

performance in versions v1 and v3. On the WN18RR dataset, GSELI achieves the highest performance across all versions, with its average performance surpassing other models. Notably, DEKG-ILP also performs well on these datasets, though it is slightly outperformed by GSELI. This indicates that although GSELI is specifically designed for bridging-inductive links, it also demonstrates excellent performance in fully-inductive link prediction tasks.

Figure 5 presents the statistical significance analysis of the v1, v2, v3, and v4 versions of the FB15K-237 and WN18RR datasets on Hits@10, comparing GSELI's performance against baseline models. For the FB15K-237 dataset, all versions have positive t-statistics and p values less than 0.015, indicating significant differences compared to baseline models, with version v4 performing best (t-statistic = 6.00, p value = 0.0018). For the WN18RR dataset, all versions also have positive t-statistics and p values less than 0.016, showing significant differences, with version v1 performing best (t-statistic = 5.46, p value = 0.0028). Overall, GSELI significantly outperforms the baseline models in all versions across both datasets.

5.5 Ablation study

This section presents the ablation study results for MRR and Hits@10 on the EQ, MB, and MF versions of the FB15k-237 dataset, focusing on fully-inductive and bridging-inductive links.

Our goal is to analyze the individual impact of each GSELI component by removing their respective contributions. These components include: (1) relation-specific semantic features (GSELI-S), (2) semantic-aware contrastive learning (GSELI-C), (3) PPR-based local clustering subgraph extraction (GSELI-L), and (4) complete neighboring relations within subgraphs (GSELI-R).

Figure 6 clearly demonstrates the superior MRR and Hits@10 performance of the original GSELI compared to its variants. Specifically:

(1) Overall, GSELI's Hits@10 outperforms all other variants across datasets. For MRR, GSELI is lower than the GSELI-R variant in FB15k-237 EQ's fully-inductive links and FB15k-237 MB's bridging-inductive links, but higher in all other cases. This indicates that while our components enhance overall performance, com-

Table 6 Percentage reduction of subgraph size for the v1, v2, v3, and v4 versions of the FB15k-237, NELL-995, and WN18RR datasets

	FB15k-237				NELL-995				WN18RR			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
GraIL	433.7	804.66	1470.68	1965.24	76.45	666.37	1427.08	1282.89	7.34	8.23	9.47	7.29
GraIL+PPR	6.85	5.16	4.05	3.58	11.45	7.45	6.14	4.12	4.46	4.44	4.73	4.46
Reduction (%)	98.42	99.36	99.72	99.82	85.02	98.88	99.57	99.68	39.24	46.05	50.05	38.82

plete neighboring relations contribute more significantly to MRR.

(2) GSEIL-S: The gap between GSELI-S and other variants highlights the importance of extracting relation-specific semantic features from the original KG. This demonstrates that rich semantic features positively impact both fully-inductive and bridging-inductive links, especially the latter.

(3) GSEIL-C: Semantic-aware contrastive learning optimizes relation-specific semantic features by generating positive and negative samples. The performance drop of GSELI-C compared to GSELI underscores this component's effectiveness.

(4) GSEIL-L: PPR-based local clustering samples tightly related subgraphs, effectively mitigating the exponential growth of neighbors within fixed hops. The improvement from GSELI-L to GSELI highlights this method's advantage.

(5) GSEIL-R: Incorporating complete neighboring relations (including features and paths) addresses the issue of sparse subgraphs. The enhancement from GSELI-R to GSELI demonstrates this approach's effectiveness.

(6) The reasons why certain modules have a smaller impact on the experiments or even outperform GSELI are as follows: these modules exhibit strong independence, resulting in minimal effect on overall performance; dataset characteristics make them particularly effective on specific datasets; parameter tuning makes them more suitable for the current task; they perform well individually but may cause conflicts or redundancy when combined with others; complex models may lead to overfitting, whereas simpler modules can better capture the essential data characteristics.

6 Further experiments

This section,"Further Experiments,"includes the following parts: subgraph extraction efficiency (see Sect. 6.1), hyper-parameter sensitivity study (see Sect. 6.2), efficiency analysis (see Sect. 6.3), and Case Study (see Sect. 6.4).

6.1 Subgraph extraction efficiency

This section validates the performance of PPR-based subgraph extraction by examining subgraph size. We compare GraIL and GraIL+PPR, as all baseline models use conventional extraction methods. Table 6 shows the percentage reduction in subgraph size for the v1, v2, v3, and v4 versions of the FB15k-237, NELL-995, and WN18RR datasets.

The table 6 reveals that GraIL+PPR significantly reduces subgraph size in all datasets and versions. Notably, the FB15k-237 and NELL-995 datasets exhibit reductions up to 99.82%. In the WN18RR dataset, reductions are smaller but still substantial, with a minimum of 38.82%. These results indicate that PPR-based subgraph extraction substantially reduces subgraph size, decreasing the nodes and edges processed during graph neural network computations. This reduction lowers computational complexity and memory usage without compromising model accuracy and effectiveness. Thus, PPR-based subgraph extraction effectively removes redundant information while retaining essential data, enhancing overall model performance.

6.2 Hyper-parameter sensitivity study

This section explores the parameter sensitivity of GSELI on different datasets. The hyperparameter θ (controlling the degree of random operations of the relations) and the parameter λ (adjusting the ratio of supervised learning to contrastive loss) are critical factors. We used the FB15K-237 and WN18RR datasets, focusing on fully-inductive and bridging-inductive links, to evaluate GSELI's Hits@10 results with parameters θ and λ ranging from 0.1 to 0.9 in increments of 0.2.

As shown in Fig. 7, our observations indicate that the Hits@10 results in the EQ of the FB15K-237 and NELL-995 datasets are influenced by parameters θ and λ. For the FB15K-237 EQ dataset, performance is generally better when $\theta \in [0.3, 0.5]$ and $\lambda \in [0.1, 0.5]$. Specifically, for fully-inductive links, the highest performance is achieved with $\theta = 0.5$ and $\lambda = 0.3$. For bridging-inductive links, the highest

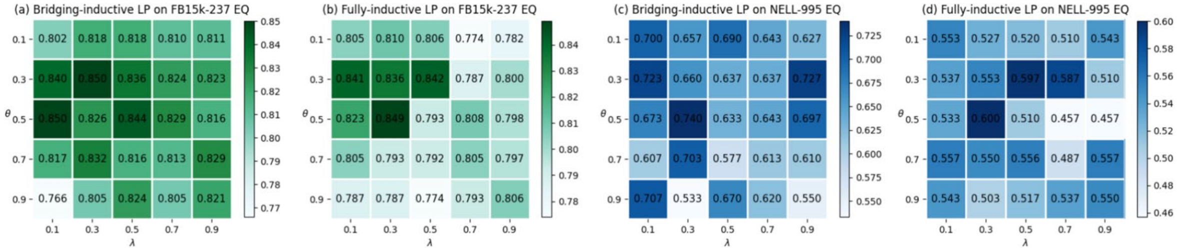

Fig. 7 The heat map of the Hit@10 results on EQ of FB15k-237 and NELL-995, for both solely fully-inductive links and solely bridging-inductive links

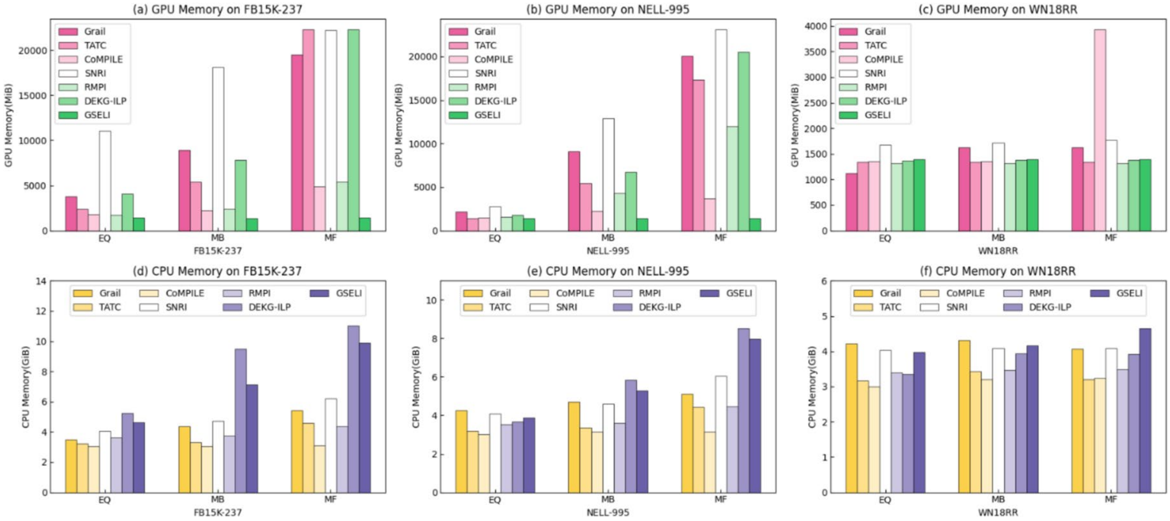

Fig. 8 GPU and CPU memory usage during the training phase for EQ, MB, and MF on FB15K-237, WN18RR, and NELL-995 datasets

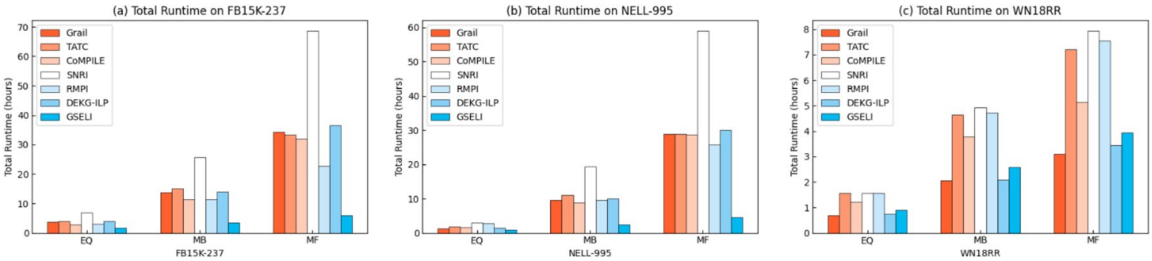

Fig. 9 Total runtime on EQ, MB, and MF for the FB15K-237, WN18RR, and NELL-995 datasets, including training and testing on mixed, bridging-inductive, and fully-inductive datasets

performance is reached with $\theta = 0.3$ and $\lambda = 0.3$, as well as $\theta = 0.5$ and $\lambda = 0.1$. Additionally, in the EQ of the NELL-995 dataset, the best performance for both fully-inductive and bridging-inductive links is achieved with $\theta = 0.5$ and $\lambda = 0.3$. Therefore, we select $\theta = 0.5$ and $\lambda = 0.3$ as the optimal parameter settings.

6.3 Efficiency analysis

This section examines the memory usage and total runtime of GSELI and baseline models on the EQ, MB, and MF versions of the FB15K-237, WN18RR, and NELL-995 datasets. All baseline models utilized both GPU and CPU during the training phase. However, only DEKG and GSELI continued using GPU and CPU during the ranking test phase, while the other models used only the CPU. To ensure a fair comparison, this paper reports memory usage only during the training phase. The total runtime includes both the training time and the testing time on mix, bridging, and fully-inductive datasets.

Figure 8 illustrates the GPU and CPU usage of GSELI and baseline models across the EQ, MB, and MF versions of the FB15K-237, NELL-995, and WN18RR datasets. The results indicate that GSELI significantly reduces GPU memory usage in most cases compared to other models. Notably, on the FB15K-237 and NELL-995 datasets, GSELI has the lowest GPU memory usage, demonstrating superior memory efficiency. Although GSELI does not show significant advantages in CPU memory usage, its consumption remains within acceptable limits. Overall, GSELI excels in GPU memory usage and maintains stability in CPU memory usage.

Figure 9 depicts the total runtime of GSELI and baseline models across the EQ, MB, and MF versions of the FB15K-237, NELL-995, and WN18RR datasets. The results show that GSELI has a significantly lower runtime for the MB and MF versions on the FB15K-237 dataset. For the NELL-995 dataset, GSELI has the shortest total runtime for the EQ and MF versions. On the WN18RR dataset, GSELI's runtime for the MB and MF versions is also significantly lower than other models. Overall, GSELI demonstrates significantly lower runtime across various datasets, showcasing higher operational efficiency.

Table 7 Rule generation from NELL-995 EQ and WN18RR EQ datasets

Rule	Score
NELL-995 EQ	
company_*economic*_sector(X, Y) ∧ agent_belongs_to_organization(Y, Z) ∧ acquired(Z, W) ∧ organization_hired_person(W, V) ⇒ works_for(X, V)	0.349
company_economic_sector(X, Y) ∧ works_for(Y, Z) ∧ acquired(Z, W) ∧ organization_hired_person(W, V) ⇒ works_for(X, V)	0.347
organization_hired_person(X, Y) ∧ agent_belongs_to_organization(Y, Z) ∧ acquired(Z, W) ∧ organization_hired_person(W, V) ⇒ works_for(X, V)	0.330
subpart_of(X, Y) ∧ company_economic_sector(Y, Z) ∧ works_for(Z, W) ∧ subpart_of(W, V) ⇒ subpart_of(X, V)	0.700
subpart_of(X, Y) ∧ subpart_of_organization(Y, Z) ∧ works_for(Z, W) ∧ subpart_of(W, V) ⇒ subpart_of(X, V)	0.698
subpart_of(X, Y) ∧ organization_headquartered_in_city(Y, Z) ∧ works_for(Z, W) ∧ subpart_of(W, V) ⇒ subpart_of(X, V)	0.695
WN18RR EQ	
hypernym(X, Y) ∧ similar_to(Y, Z) ∧ similar_to(Z, W) ∧ similar_to(W, V) ⇒ has_part(X, V)	0.973
similar_to(X, Y) ∧ similar_to(Y, Z) ∧ instance_hypernym(Z, W) ∧ member_meronym(W, V) ⇒ has_part(X, V)	0.970
similar_to(X, Y) ∧ similar_to(Y, Z) ∧ has_part(Z, W) ∧ similar_to(W, V) ⇒ has_part(X, V)	0.969
verb_group(X, Y) ∧ similar_to(Y, Z) ∧ similar_to(Z, W) ∧ synset_domain_topic_of(W, V) ⇒ also_see(X, V)	0.993
also_see(X, Y) ∧ also_see(Y, Z) ∧ instance_hypernym(Z, W) ∧ has_part(W, V) ⇒ also_see(X, V)	0.992
verb_group(X, Y) ∧ synset_domain_topic_of(Y, Z) ∧ similar_to(Z, W) ∧ synset_domain_topic_of(W, V) ⇒ also_see(X, V)	0.992

Additionally, Table 2 reveals that the number of the EQ, MB, and MF versions of the FB15K-237, NELL-995, and WN18RR datasets gradually improves. However, the results from Figs. 8 and 9 show that GSELI maintains high efficiency in GPU memory usage and total runtime. This further demonstrates GSELI's ability to remain efficient and scalable when handling large-scale datasets, giving it an advantage in large-scale data processing scenarios.

6.4 Case study

Subgraphs, as effective path combinations between target entities, offer more comprehensive insights than single rules, revealing the rationale of each target link (u, r_t, v) within the cycle. We generated all relation rule cycles of lengths up to four and input them into the GSELI model for scoring. These rules include the target relation (rule head) and the relation path (rule body). Using the sigmoid function, we normalized the scores and selected the top three cycles with the highest scores.

Table 7 presents examples of EQ rule generation by GSELI on the NELL-995 and WN18RR datasets, highlighting its explanatory power and applicability to complex datasets. GSELI identifies and interprets complex relations through rule cycles, generating high-scoring paths for more meaningful link explanations. This enhances the model's prediction accuracy, transparency, and interpretability, demonstrating GSELI's robust capability and potential in handling diverse and complex datasets.

7 Conclusion

This paper presents GSELI, an ILP model that integrates global semantic features and enhanced local subgraphs. It enables predictions for both unseen-unseen entities in emerging KGs (fully-inductive links) and the more challenging unseen entities from the original KG to the emerging KG (bridging-inductive links). GSELI extracts CL-based global relation semantic features, effectively addressing the topological limitations of both the original KG and the emerging KG. To further enhance the topological information of subgraphs, GSELI employs a PPR-based local clustering approach to sample tightly-related subgraphs, mitigating the exponential growth of neighbors within a fixed number of hops. Additionally, it integrates complete neighboring relations to address sparse subgraphs. Experiments show GSELI's superior performance compared to state-of-the-art models for both fully-inductive and bridging-inductive links.

Our future research plans include: (1) conducting a comprehensive evaluation of fully-inductive and bridging-inductive links; (2) advancing more challenging ILP involving unseen entities and relations; (3) leveraging large language models to enhance ILP.

Acknowledgements This research is supported by National Natural Science Foundation of China (Grant Number: 61375084), the Shandong Natural Science Foundation of China (Grant Number: ZR2019MF064), and the Technological Small and Medium-sized Enterprise Innovation Ability Enhancement Project (Grant Number: 2022TSGC2189). Thanks are due to all the anonymous reviewers.

Data availability The datasets supporting the results of this article are included within the references.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Huang X, Zhang J, Li D et al (2019) Knowledge graph embedding based question answering. In: Proceedings of the twelfth ACM international conference on web search and data mining, pp 105–113
2. Wang X, Wang D, Xu C et al (2019) Explainable reasoning over knowledge graphs for recommendation. In: Proceedings of the AAAI conference on artificial intelligence, pp 5329–5336
3. Xiong C, Power R, Callan J (2017) Explicit semantic ranking for academic search via knowledge graph embedding. In: Proceedings of the 26th international conference on world wide web, pp 1271–1279
4. Dong X, Gabrilovich E, Heitz G et al (2014) Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 601–610
5. Xie R, Liu Z, Jia J et al (2016) Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the AAAI conference on artificial intelligence, pp 2659–2665
6. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Advances in neural information processing systems, vol 30
7. Zeng H, Zhou H, Srivastava A et al (2019) Graphsaint: graph sampling based inductive learning method. arXiv:1907.04931
8. Hamaguchi T, Oiwa H, Shimbo M et al (2017) Knowledge transfer for out-of-knowledge-base entities: a graph neural network approach. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp 1802–1808
9. Bi Z, Zhang T, Zhou P et al (2020) Knowledge transfer for out-of-knowledge-base entities: improving graph-neural-network-based embedding using convolutional layers. IEEE Access 8:159039–159049
10. Wang C, Zhou X, Pan S et al (2022) Exploring relational semantics for inductive knowledge graph completion. In: Proceedings of the AAAI conference on artificial intelligence, pp 4184–4192
11. Teru K, Denis E, Hamilton W (2020) Inductive relation prediction by subgraph reasoning. In: International conference on machine learning, pp 9448–9457
12. Chen J, He H, Wu F et al (2021) Topology-aware correlations between relations for inductive link prediction in knowledge graphs. In: Proceedings of the AAAI conference on artificial intelligence, pp 6271–6278
13. Mai S, Zheng S, Yang Y et al (2021) Communicative message passing for inductive relation reasoning. In: Proceedings of the AAAI conference on artificial intelligence, pp 4294–4302
14. Xu X, Zhang P, He Y et al (2022) Subgraph neighboring relations infomax for inductive link prediction on knowledge graphs. arXiv:2208.00850
15. Mohamed HA, Pilutti D, James S et al (2023) Locality-aware subgraphs for inductive link prediction in knowledge graphs. Pattern Recogn Lett 167:90–97
16. Geng Y, Chen J, Pan JZ et al (2023) Relational message passing for fully inductive knowledge graph completion. In: Proceedings of the 39th international conference on data engineering (ICDE), pp 1221–1233
17. Ali M, Berrendorf M, Galkin M et al (2021) Improving inductive link prediction using hyper-relational facts. In: Proceedings of the 20th international semantic web conference, pp 74–92
18. Zhang Y, Wang W, Yin H et al (2023) Disconnected emerging knowledge graph oriented inductive link prediction. In: Proceedings of the 39th international conference on data engineering (ICDE), pp 381–393
19. Bordes A, Usunier N, Garcia-Duran A et al (2013) Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems, vol 26
20. Wang Z, Zhang J, Feng J et al (2014) Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI conference on artificial intelligence, pp 1112–1119
21. Nickel M, Tresp V, Kriegel HP et al (2011) A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th international conference on machine learning, pp 3104482–3104584
22. Yang B, Yih Wt, He X et al (2014) Embedding entities and relations for learning and inference in knowledge bases. arXiv:1412.6575
23. Schlichtkrull M, Kipf TN, Bloem P et al (2018) Modeling relational data with graph convolutional networks. In: Proceedings of the 15th international semantic web conference, pp 593–607
24. Vashishth S, Sanyal S, Nitin V et al (2019) Composition-based multi-relational graph convolutional networks. arXiv:1911.03082
25. Galárraga LA, Teflioudi C, Hose K et al (2013) Amie: association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of the 22nd international conference on world wide web, pp 413–422
26. Meilicke C, Fink M, Wang Y et al (2018) Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion. In: Proceedings of the 17th international semantic web conference, pp 3–20
27. Meilicke C, Chekol MW, Ruffinelli D et al (2019) Anytime bottom-up rule learning for knowledge graph completion. In: Proceedings of the 28th international joint conference on artificial intelligence, pp 3137–3143
28. Yang F, Yang Z, Cohen WW (2017) Differentiable learning of logical rules for knowledge base reasoning. In: Advances in neural information processing systems, vol 30
29. Sadeghian A, Armandpour M, Ding P et al (2019) Drum: end-to-end differentiable rule mining on knowledge graphs. In: Advances in Neural Information Processing Systems, vol 32
30. Qu M, Chen J, Xhonneux LP et al (2020) Rnnlogic: learning logic rules for reasoning on knowledge graphs. arXiv:2010.04029
31. Wang L, Zhao W, Wei Z, et al (2022) Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 4281–4294
32. Pan Y, Liu J, Zhang L et al (2021) Learning first-order rules with relational path contrast for inductive relation reasoning. arXiv:2110.08810
33. Kwak H, Jung HBK (2022) Subgraph representation learning with hard negative samples for inductive link prediction. In: Proceedings of the 2022 international conference on acoustics, speech and signal processing (ICASSP), pp 4768–4772
34. Gilmer J, Schoenholz SS, Riley PF et al (2017) Neural message passing for quantum chemistry. In: International conference on machine learning, pp 1263–1272
35. Cho K, Bart, Bahdanau D et al (2014) Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1724–1734

36. Toutanova K, Chen D, Pantel P et al (2015) Representing text for joint embedding of text and knowledge bases. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 1499–1509
37. Xiong W, Hoang T, Wang WY (2017) Deeppath: a reinforcement learning method for knowledge graph reasoning. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 564–573
38. Dettmers T, Minervini P, Stenetorp P et al (2018) Convolutional 2d knowledge graph embeddings. In: Proceedings of the AAAI conference on artificial intelligence, pp 1811–1818
39. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.