Structure-Aware Zero-Shot Relational Learning for Knowledge Graphs without External Knowledge

Kuan Xu[1,2], Baoxin Zhang[1,2], Shuyue Fan[1,2], Ming Chen[3], Zhipeng Ke[4], Jian Yu[1,2], Xuezhong Zhou[*1,2],

[1]School of Computer Science & Technology, Beijing Jiaotong University, Beijing, China,
[2]Beijing Key Laboratory of Traffic Data Mining and Embodied Intelligence, Beijing, China,
[3]China Mobile Financial Technology Co., Ltd., Beijing, China,
[4]State Key Laboratory of Technologies for Chinese Medicine Pharmaceutical Process Control and Intelligent Manufacture, Nanjing, Jiangsu, China.

{xukuan, baoxin, shuyuefan, jianyu, xzzhou}@bjtu.edu.cn, kzp0901@163.com, chenmingds@chinamobile.com

Abstract

Zero-shot Relational Learning (ZRL) aims to perform knowledge graph completion when dealing with newly emerging relations that have no observed triples. However, existing ZRL methods typically rely on external knowledge (e.g., textual descriptions or curated annotations), which increases labeling costs and limits its real-world applicability. To overcome this limitation, we propose a new **S**tructure-**A**ware paradigm for **ZRL**, termed **SAZRL**, that performs ZRL without relying on external knowledge. SAZRL leverages intrinsic structural patterns in KGs to bridge semantic correlations for new relations with existing ones. It constructs structure-aware conditional query graphs based on shared entities and adaptive relation updating module to infer representations for new relations based on the query graphs. We conduct extensive experiments on three real-world benchmarks, **NELL-ZS**, **Wiki-ZS** and **FB15K-ZS**, demonstrating that SAZRL consistently surpasses state-of-the-art ZRL methods, achieving up to **10.66%** improvement in MRR while eliminating external annotations.

1 Introduction

Knowledge Graphs (KGs), constructed from factual facts (i.e., (h, r, t)), play a vital role in various tasks, e.g., drug repurposing (Kumar et al., 2025), recommendation systems (Cui et al., 2025), and information retrieval (Rossi et al., 2021). Despite recent advances, most existing KGs remain largely incomplete (Li et al., 2025; Meilicke et al., 2024). To address this issue, Knowledge Graph Completion (KGC) has emerged as a crucial task that aims to predict missing entities—either the tail t given a head–relation pair (h, r) or the head h given a relation–tail pair (r, t) (Chen et al., 2024).

Recent studies (Xu et al., 2025; Chen et al., 2024) have made great progress in KGC. However, these methods struggle with newly emerging relations. Although collecting new instances and retraining models can alleviate this issue, such solutions are often impractical due to the high cost of annotation and retraining (Geng et al., 2021). To address this challenge, zero-shot relation learning (ZRL) (Qin et al., 2020; Dong et al., 2024) has emerged, which aims to infer facts involving new relations without requiring their instances.

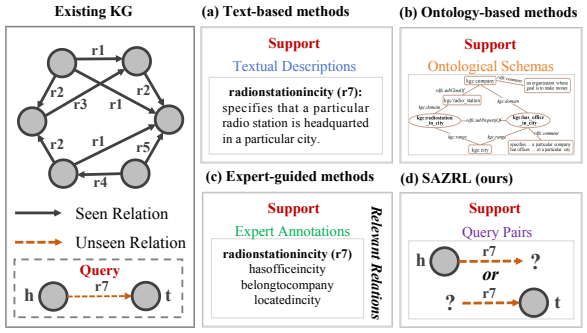


Figure 1: Various ZRL methods. Ours (**d**) does not rely on external knowledge compared with others.

The key idea of ZRL lies in acquiring prior knowledge (i.e., *support set*) for new (i.e., *unseen*) relations to model semantic correlations with existing (i.e., *seen*) ones within the prior knowledge space, thereby leveraging recognition experience (i.e., *model parameters*) learned from seen relations to make prediction on unseen ones (i.e., *query set*) (Chen et al., 2023). For example, as shown in Figure 1, text-based methods (Qin et al., 2020; Li et al., 2024; Fang et al., 2025) and ontology-based methods (Geng et al., 2021, 2022) utilize textual descriptions or ontological schemas to capture semantic relationships between relations in the textual or ontological space. Expert-guided methods (Dong et al., 2024), on the other hand, rely on manually annotated relevance scores between relations derived from expert knowledge. Although these methods enable predictions without training instances for unseen relations, such external knowledge is not always readily available.

Findings of the Association for Computational Linguistics: ACL 2026, pages 18855–18869
July 2-7, 2026

Therefore, we wonder if there exists an approach that can conduct ZRL without relying on external knowledge (e.g., **textual descriptions**, **ontological schemas**, or **expert annotations**).

In this paper, we propose such a solution. Our goal is to encode semantic correlations among relations through the structural patterns within KGs , as semantically related relations typically share same entities (Galkin et al., 2024; Lee et al., 2023).

To this end, we propose **S**tructure-**A**ware **ZRL** (**SAZRL**), a new paradigm that exploits structural patterns to enable effective ZRL without relying on external knowledge. Specifically, SAZRL consists of two core components: (1) the **C**onditional **Q**uery **G**raphs (**CQG**) module and (2) the **A**daptive **R**elation **U**pdating (**ARU**) module. In **CQG**, we utilize the query pairs with unseen relation (e.g., $(h, r^{unseen}, ?)$ or $(?, r^{unseen}, t)$) to extract subgraph conditioned on them, which aims to identify relations semantically similar to the query relation. Then, the query graph for relation r^{unseen} can be constructed by leveraging shared entities (e.g., h or t) between query pairs and known facts from the subgraph. In this graph, each node represents a relation, and each weighted edge encodes the semantic similarity between two relations. The similarity between two relations is measured by considering the entities they share. Based on this graph, the relevant neighboring relations for unseen relation r^{unseen} are identified. The **ARU** then synthesizes embeddings for r^{unseen} by aggregating the embeddings of its neighboring seen relations via an adaptive attention mechanism. The process is optimized by maximizing the log-likelihood of observed triples based on relation-entity interaction plausibility scores.

To ensure a fair comparison, we evaluate SAZRL against state-of-the-art ZRL methods on three benchmarks: **NELL-ZS** and **Wiki-ZS**, which provide rich external knowledge for ZRL, and **FB15K-ZS**, a new dataset derived from FB15K-237 (Toutanova and Chen, 2015), which does not include external knowledge. Extensive experiments across all datasets show that SAZRL consistently outperforms existing methods. Notably, SAZRL requires no external knowledge beyond the KG, demonstrating its advantages in reducing annotation costs and enhancing practicality. Our main contributions are as follows:

- We propose a new **S**tructure-**A**ware **Z**ero-shot **R**elational **L**earning (**SAZRL**) paradigm which exploits structural patterns in the KG to perform KGC for unseen relations.
- Unlike existing methods, SAZRL enables ZRL without relying on external knowledge, reducing annotation costs and enhancing practicality in real-world scenarios.
- We experiment on three datasets: **NELL-ZS**, **Wiki-ZS**, and **FB15K-ZS**. SAZRL consistently outperforms existing ZRL methods, achieving **10.66%** and **4.33%** improvements in MRR on NELL-ZS and Wiki-ZS, and performs effectively on FB15K-ZS.

2 Related Work

Existing studies on ZRL can be broadly categorized into three groups: text-based methods, ontology-based methods, and expert-guided methods.

Text-based Methods. ZSGAN (Qin et al., 2020) employs generative adversarial networks (GANs) to encode textual descriptions of unseen relations and generate plausible embeddings for them. CZRL (Fang et al., 2025) introduces a contrastive learning framework that differentiates relation representations derived from textual descriptions, thereby refining model performance. ROCKGC (Yu and Yu, 2025) leverages LLMs to enrich the textual descriptions of relations.

Ontology-based Methods. Ontology-based approaches leverage structured ontological knowledge. OntoZSL (Geng et al., 2021) generates embeddings for unseen relations based on an ontological schema, while DOZSL (Geng et al., 2022) further disentangles relation representations within the same schema, producing more fine-grained embeddings for unseen relations.

Expert-guided Methods. FZR (Dong et al., 2024) advances beyond text-based approaches by identifying the underlying semantic factors in textual descriptions through factor extraction and shared-factor composition. In addition, it introduces a human-expert–guided mechanism to reconstruct relation representations, where experts provide relevance scores between relation pairs, thus enhancing semantic fidelity.

Despite their effectiveness, they are highly dependent on external knowledge, which may be unavailable or costly in practical settings. In contrast, our work investigates structure-aware ZRL, which operates without relying on such knowledge.

Beyond the ZRL, we also note the recent rise of fully inductive KGC task, which similarly involves reasoning over unseen relations. We provide a detailed discussion of it and clarify the fundamental distinctions from ZRL in **Appendix** A.1 and A.2 .

3 Preliminaries

In this section, we provide a formal definition of the problem of ZRL and outline the key notations used throughout this paper.

In the context of ZRL, a KG can be formally represented as $\mathcal{K} = (\mathcal{E}, \mathcal{R}_s, \mathcal{R}_u, \mathcal{T}_s, \mathcal{T}_u)$, where $\mathcal{E}$ denotes the set of entities, $\mathcal{R}_s$ and $\mathcal{R}_u$ denote the sets of seen and unseen relations, respectively. Correspondingly, $\mathcal{T}_s$ and $\mathcal{T}_u$ represent the sets of triplets involving seen and unseen relations i.e.,

$$\mathcal{T}_s = \{(e_i, r_k^s, e_j) | e_i, e_j \in \mathcal{E}, r_k^s \in \mathcal{R}_s\},$$
$$\mathcal{T}_u = \{(e_i, r_k^u, e_j) | e_i, e_j \in \mathcal{E}, r_k^u \in \mathcal{R}_u\}.$$

Note that $\mathcal{R}_s \cap \mathcal{R}_u = \emptyset$, indicating that the relations appearing in $\mathcal{R}_u$ do not appear in $\mathcal{R}_s$, and vice versa. According to standard ZRL settings, as adopted in prior ZRL studies (Geng et al., 2022; Dong et al., 2024), a closed-world setting for entities is considered, meaning that each entity appearing in $\mathcal{T}_u$ also exists in $\mathcal{T}_s$. The objective of ZRL is formulated as predicting the missing entity given a query pair which consists of a query entity and a unseen query relation. For example, for the query pair (e_i, r_k^u), there exists a ground-truth tail entity e_j and a corresponding candidate set $\mathcal{C}_{(e_i, r_k^u)}$. Our goal is to assign the highest ranking to e_j among the candidate entities $e_j' \in \mathcal{C}_{(e_i, r_k^u)}$.

During training, the set of seen triplets $\mathcal{T}_s$ is partitioned into two subsets. The first subset, denoted as $\mathcal{T}_g = \{(e_i, r_k^{ss}, e_j) \mid e_i, e_j \in \mathcal{E},\ r_k^{ss} \in \mathcal{R}_{ss}\}$, serves as a background KG and provides the triplets involving seen relations. The second subset, denoted as $\mathcal{T}_{tr} = \{(e_i, r_k^{su}, e_j) \mid e_i, e_j \in \mathcal{E},\ r_k^{su} \in \mathcal{R}_{su}\}$, is used for training and simulates the triplets involving unseen relations. Here, $\mathcal{R}_{ss} \subset \mathcal{R}_s$, $\mathcal{R}_{su} \subset \mathcal{R}_s$, and $\mathcal{R}_{ss} \cap \mathcal{R}_{su} = \emptyset$. Meanwhile, the set of unseen triplets $\mathcal{T}_u$ is further divided into a validation set $\mathcal{T}_{val}$ and a testing set $\mathcal{T}_{test}$ to evaluate the performance of the proposed method.

It is worth noting that during the training phase, $\mathcal{T}_{tr}$ is dynamically sampled from $\mathcal{T}_s$, with the remaining triplets used as $\mathcal{T}_g$. In other words, $\mathcal{T}_{tr}$ and $\mathcal{T}_g$ are not fixed throughout training. For clarity of exposition, we present our method assuming a fixed partition in the following descriptions. Additionally, during the validation and testing phases, the entire $\mathcal{T}_s$ is used as the background KG.

4 The Proposed Method

In this section, we present a detailed description of our proposed method, SAZRL. The detailed *Algorithm* of training and testing is provided in **Appendix** B. SAZRL consists of three components:

1. the structure-aware **C**onditional **Q**uery **G**raph (**CQG**) construction module (Figure 2(a)-(c)): Constructs conditional query graphs based on query pairs to establish correlations between seen and unseen relations by leveraging structural patterns in the KG;

2. the **A**daptive **R**elation **U**pdating (**ARU**) module (Figure 2(d)): Updates relation representations by aggregating neighboring seen relations through attention index to generate embeddings for unseen ones;

3. the Relation-Entity Interaction Modeling module (Figure 2 (e)): Models relation–entity interactions and optimizes the model through an interaction scoring function.

4.1 Structure-Aware CQG Construction

We decompose the CQG construction process into two components: (1) **conditional subgraph extraction** (Figure 2 (a-b)), and (2) **query graph construction** (Figure 2 (c)).

The goal of conditional subgraph extraction is to identify relations semantically similar to the query relation, which typically share entities and exhibit clustering (Ding et al., 2025; Lee et al., 2023).

To begin with, during training, each triplet in $\mathcal{T}_{tr}$ is decomposed into two binary query pairs. For example, the triplet (e_i, r_k^{su}, e_j) is split into (e_i, r_k^{su}) and (r_k^{su}, e_j). **Note that only a single query pair (e.g., $(e_i, r_k^u, ?)$) is used during testing to prevent data leakage.** Based on these two query pairs, we extract a subgraph $\mathcal{S}_{sub}^{r_k^{su}}$ from the background graph $\mathcal{T}_g$. Specifically, we extract the entities e_i and e_j from the query pairs (e_i, r_k^{su}) and (r_k^{su}, e_j) respectively, to construct an entity set denoted as $\mathcal{E}_q^{r_k^{su}}$. Since all entities in $\mathcal{E}_q^{r_k^{su}}$ are observable in $\mathcal{T}_g$, we uniformly sample the two-hop neighbors for each entity in $\mathcal{E}_q^{r_k^{su}}$ from $\mathcal{T}_g$ to construct the entity set $\mathcal{E}_{sub}^{r_k^{su}}$. To avoid exponential expansion, we limit

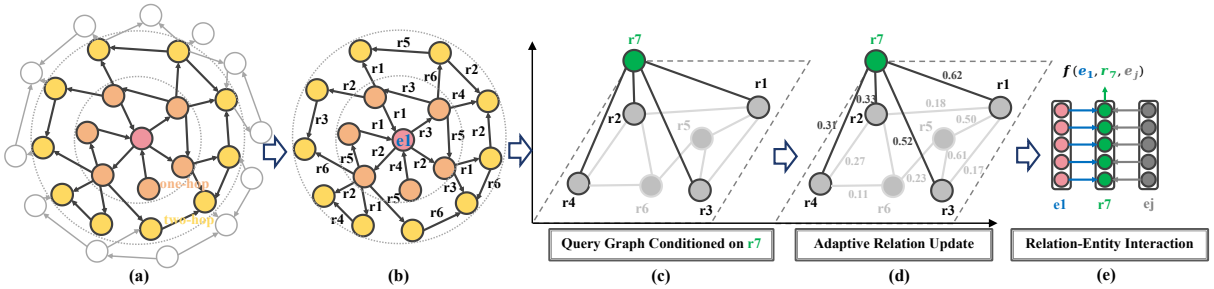


Figure 2: Overview of SAZRL with the query pair $(e_1, r_7, ?)$ as an example. A weighted query graph is built, where e_1 bridges the **unseen relation** r_7 and **seen relations** r_1-r_6, allowing r_7 to be inferred from its neighbors.

the number of neighbors to at most 50 per hop in line with common practice. Then, the subgraph conditioned on relation r_k^{su} is defined as:

$$\mathcal{S}_{sub}^{r_k^{su}} = \{(e_i, r_k^{ss}, e_j) | e_i, e_j \in \mathcal{E}_{sub}^{r_k^{su}}, r_k^{ss} \in \mathcal{R}_{ss}\}. \quad (1)$$

Given the subgraph $\mathcal{S}_{sub}^{r_k^{su}}$ and the query pairs extracted from $\mathcal{T}_{tr}$, we build a relation-conditioned query graph $\mathcal{G}^{r_k^{su}}$ as a weighted graph. In $\mathcal{G}^{r_k^{su}}$, each node represents a relation, and each edge weight encodes the semantic similarity between the corresponding pair of relations. Intuitively, multiple similarity measures can be used to encode semantic relatedness, e.g., **co-occurrence**, **jaccard similarity**, and **positive pointwise mutual information**. For clarity, we present the co-occurrence-based similarity formulation here. The other two similarity measures, along with their comparative experimental results, are provided in **Appendix** C.

To construct $\mathcal{G}^{r_k^{su}}$, we first define two matrices:

$$M_h \in \mathbb{R}^{n \times (m+m')}, \quad M_t \in \mathbb{R}^{n \times (m+m')}, \quad (2)$$

where n denotes the number of entities in the subgraph $\mathcal{S}_{sub}^{r_k^{su}}$, m is the number of relations in $\mathcal{S}_{sub}^{r_k^{su}}$, and m' is the number of relations derived from query pairs. The matrices M_h and M_t encode the frequency with which each entity appears as a head or tail entity, respectively. For instance, $M_h[i, j]$ indicates the number of times entity e_i appears as the head entity of relation r_j. Next, we compute the relation similarity matrices as:

$$\mathcal{G}_h^{r_k^{su}} = M_h^T D_h^{-2} M_h, \quad \mathcal{G}_t^{r_k^{su}} = M_t^T D_t^{-2} M_t, \quad (3)$$

where the $D_h, D_t \in \mathbb{R}^{n \times n}$ denote diagonal matrices representing the degrees of entities when they act as head and tail, respectively. Specifically, $D_h[i, i] = \sum_j M_h[i, j]$ and $D_t[i, i] = \sum_j M_t[i, j]$. The use of D_h^{-2} and D_t^{-2} ensures that the similarity weights in $\mathcal{G}_h^{r_k^{su}}$ and $\mathcal{G}_t^{r_k^{su}}$ are normalized to sum to one, which helps mitigate the impact of imbalanced entity participation across relations. Finally, the query graph $\mathcal{G}^{r_k^{su}}$ is represented as an adjacency matrix, defined as follows:

$$\mathcal{G}^{r_k^{su}} = \mathcal{G}_h^{r_k^{su}} + \mathcal{G}_t^{r_k^{su}}, \quad \mathcal{G}^{r_k^{su}} \in \mathbb{R}^{(m+m') \times (m+m')}, \quad (4)$$

where each element $s_{ij} \in \mathcal{G}^{r_k^{su}}$ denotes the semantic similarity between relations r_i and r_j. This formulation enables the modeling of correlations between seen and unseen relations without relying on external knowledge beyond KGs. Note that the query graph is not designed to identify an "optimal" set of similar relations; rather, it defines a reasonable neighborhood for the target unseen relation.

4.2 ARU: Adaptive Relation Updating

For each relation r_k $(k = 1, 2, \ldots, m+m')$ in $\mathcal{G}^{r_k^{su}}$, we associate a feature vector $x_k \in \mathbb{R}^d$, where d is the feature dimension. Following prior work (Yang et al., 2024), we initialize x_k using Glorot initialization (Glorot and Bengio, 2010). We define the hidden representation of each relation r_k at layer l as $h_k^l \in \mathbb{R}^{d'}$, where d' is the dimensionality of the hidden space. The superscript l indicates the l-th layer in the network, with $l = 0, \ldots, L-1$. For each relation r_k, the hidden representation at layer $l+1$ is computed by aggregating the weighted hidden representations of its neighbors and itself from layer l. We also incorporate residual connections at each layer to effectively preserve and leverage the hidden representations across layers. The update rule is formally defined as:

$$h_k^{l+1} = \sigma_1\left(\left(\sum_{r_j \in N_k} \alpha_{kj}^l W^l h_j^l\right) + K^l h_k^l\right). \quad (5)$$

Here, $W^l, K^l \in \mathbb{R}^{d' \times d'}$ are learnable parameter matrices at layer l. The set N_k denotes the neighbors of relation r_k in $\mathcal{G}$. The function $\sigma_1(\cdot)$ denotes the activation function which is implemented as

ReLU (Glorot et al., 2011) in our method. The initial hidden representation at layer 0 is computed as $h_k^0 = W' x_k$, where $W' \in \mathbb{R}^{d' \times d}$ is a learnable projection matrix. α_{kj}^l denotes the attention weight between relations r_k and r_j, which is formally defined as follows:

$$\alpha_{kj}^l = \frac{\exp\left(\varpi^l \sigma_2(c_{kj}^l) + o_{I(k,j)}^l\right)}{\sum_{r_{j'} \in \mathcal{N}_k} \exp\left(\varpi^l \sigma_2(c_{kj'}^l) + o_{I(k,j')}^l\right)} \quad (6)$$

where the activation function $\sigma_2(\cdot)$ is set to LeakyReLU (Maas et al., 2013). A learnable row vector $\varpi^l \in \mathbb{R}^{1 \times d'}$ is applied after $\sigma_2(\cdot)$ to mitigate the static attention problem, as discussed in (Brody et al., 2021). The term c_{kj}^l is introduced to model semantic interactions between relations r_k and r_j. Additionally, following (Lee et al., 2023), we incorporate a global similarity bias $o_{I(k,j)}^l$, which serves as a learnable scalar determined by the discretized ranking of the similarity score s_{kj} within the query graph $\mathcal{G}^{r_k^{su}}$. Specifically, we define

$$c_{kj}^l = U^l[h_k^l || h_j^l], \quad I(k,j) = \left\lceil \frac{\text{rank}(s_{kj}) \times B}{|\mathcal{G}|_{\neq 0}} \right\rceil \quad (7)$$

where $U^l \in \mathbb{R}^{d' \times 2d'}$ is a learnable parameter matrix used to capture semantic information between r_k and r_j. The symbol $||$ denotes the concatenation operation. $\text{rank}(s_{kj})$ denotes the rank position of s_{kj} among all non-zero elements in $\mathcal{G}$ sorted in descending order. $|\mathcal{G}|_{\neq 0}$ denotes the number of non-zero elements in $\mathcal{G}$. The similarity scores are discretized into B bins, with each bin index $I(k,j) \in \{1, \ldots, B\}$ corresponds to a learnable parameter $o_{I(k,j)}^l$, selected from the set $\{o_1^l, \ldots, o_B^l\}$. Finally, the output embedding z_k of relation $r_k (k = 1, 2, \ldots, m + m')$ is defined as: $z_k = P h_k^L$, where $P \in \mathbb{R}^{\hat{d} \times d'}$ is a learnable projection matrix that maps the final hidden representation to a output space of dimension $\hat{d}$.

4.3 Relation-Entity Interaction Modeling

Given a closed set of entities, we define each entity's embedding as $v_t \in \mathbb{R}^{\hat{d}}$ $(t = 1, ..., n')$, where n' is the number of entities in $\mathcal{T}_g$ and v_t is learnable. With the embedding z_k for each relation r_k, we can model the relation-entity interaction using various interaction methods (e.g., **translation**, **product** and **rotation**). For clarity, we follow previous ZRL studies (Qin et al., 2020; Dong et al., 2024) and define our interaction function as:

$$f(e_i, r_k, e_j) = v_i^T \, diag(z_k) v_j, \quad (8)$$

where $diag(z_k)$ is a diagonal matrix whose diagonal elements are given by z_k. With the scoring function, following (Ge et al., 2023), we derive a Log-Likelihood objective function with self-adversarial negative sampling, formulated as:

$$\mathcal{L} = -\log \sigma_3\left(f(e_i, r_k, e_j)\right) \quad (9)$$
$$- \sum_{tr_{i'kj} \in \mathcal{S}_h} p(tr_{i'kj} | \{(e_i, r_k, e_j)\}) \log \sigma_3\left(-f(tr_{i'kj})\right)$$
$$- \sum_{tr_{ikj'} \in \mathcal{S}_t} p(tr_{ikj'} | \{(e_i, r_k, e_j)\}) \log \sigma_3\left(-f(tr_{ikj'})\right),$$

where $\sigma_3(\cdot)$ denotes the sigmoid function. For convenience, we denote the triplets $(e_{i'}, r_k, e_j)$ and $(e_i, r_k, e_{j'})$ as $tr_{i'kj}$ and $tr_{ikj'}$, respectively. $tr_{i'kj}$ and $tr_{ikj'}$ are negative triplets sampled from the negative sets $\mathcal{S}_h$ and $\mathcal{S}_t$, generated by replacing the head or tail entities of (e_i, r_k, e_j). $p(tr_{i'kj} | \{(e_i, r_k, e_j)\})$ and $p(tr_{ikj'} | \{(e_i, r_k, e_j)\})$ are the probabilities of sampling $tr_{i'kj}$ and $tr_{ikj'}$, respectively. For example, given a positive triplet (e_i, r_k, e_j), the sampling probability of $tr_{ikj'}$ is defined as:

$$p(tr_{ikj'} | \{(e_i, r_k, e_j)\}) = \frac{A_{j'}}{\sum_{i'} A_{i'} + \sum_{j'} A_{j'}}, \quad (10)$$

$$A_{i'} = exp(\tau f(tr_{i'kj})), A_{j'} = exp(\tau f(tr_{ikj'})), \quad (11)$$

where τ is the temperature hyperparameter of sampling, which is set to 0.5 in our experiments.

5 Experiments

5.1 Datasets and Evaluation Metrics

To the best of our knowledge, there are only two publicly available benchmarks for ZRL, namely NELL-ZS and Wiki-ZS, both introduced by (Qin et al., 2020) and adopted by subsequent work (Dong et al., 2024; Fang et al., 2025). These benchmarks are derived from real-world KGs: NELL[1] and Wikidata[2] , respectively. Both datasets provide an ontological schema over relations, where each relation is accompanied by textual descriptions and expert annotations. To further evaluate the performance of mothods in the absence of external knowledge, we further construct FB15K-ZS from FB15K-237 (Toutanova and Chen, 2015) following (Qin et al., 2020). For each dataset, the triplets $\mathcal{T}_u$ is divided into validation set $\mathcal{T}_{val}$ and test set $\mathcal{T}_{test}$, with relation sets denoted as $\mathcal{R}_{val}$ and $\mathcal{R}_{test}$, respectively. Table 2 lists the dataset statistics using the notion defined in section 3.

[1]NELL: http://rtw.ml.cmu.edu/rtw/
[2]Wikidata: https://www.wikidata.org/

Table 1: Performance comparison of ZRL methods on NELL-ZS and Wiki-ZS datasets. The optimal results are denoted in **bold**, while the sub-optimal results are underlined. "–" indicates the model is unavailable on the dataset.

Condition			Method	NELL-ZS				Wiki-ZS				FB15K-ZS			
w/o Text	w/o Onto	w/o Exp		MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1
✗	✓	✓	ZS-TransE	0.097	0.203	0.147	0.043	0.053	0.119	0.081	0.018	0.015	0.027	0.011	0.003
✗	✓	✓	ZS-DistMult	0.235	0.326	0.284	0.185	0.189	0.236	0.210	0.161	0.028	0.053	0.025	0.013
✗	✓	✓	ZS-ComplEx	0.216	0.316	0.267	0.160	0.118	0.180	0.144	0.083	0.013	0.020	0.009	0.007
✗	✓	✓	ZSGAN	0.244	0.371	0.308	0.175	0.193	0.277	0.231	0.157	-	-	-	-
✗	✗	✓	OntoZSL	0.243	0.362	0.303	0.178	0.187	0.276	0.230	0.149	-	-	-	-
✓	✗	✓	DisenE+GAN	0.219	0.351	0.288	0.155	0.190	0.272	0.230	0.138	0.031	0.067	0.027	0.010
✓	✗	✓	DisenE+GCN	0.230	0.361	0.292	0.152	0.187	0.262	0.221	0.137				
✓	✗	✓	RGAT+GAN	0.222	0.341	0.290	0.162	0.200	0.280	0.219	0.144	0.027	0.038	0.014	0.008
✓	✗	✓	RGAT+GCN	0.226	0.360	0.289	0.160	0.179	0.268	0.220	0.136				
✓	✗	✓	DisenKGAT+GAN	0.232	0.359	0.279	0.161	0.190	0.279	0.234	0.141	0.033	0.063	0.017	0.012
✓	✗	✓	DisenKGAT+GCN	0.230	0.351	0.273	0.165	0.186	0.275	0.227	0.135				
✓	✗	✓	DOZSL(RD+GAN)	0.234	0.371	0.296	0.171	0.188	0.279	0.230	0.137	-	-	-	-
✓	✗	✓	DOZSL(AGG+GAN)	0.227	0.363	0.289	0.165	0.186	0.272	0.222	0.138	-	-	-	-
✓	✗	✓	DOZSL(RD+GCN)	0.237	0.375	0.309	0.161	0.177	0.267	0.220	0.138	-	-	-	-
✓	✗	✓	DOZSL(AGG+GCN)	0.225	0.358	0.300	0.164	0.191	0.275	0.227	0.135	-	-	-	-
✗	✓	✗	FZR$_{TransE}$	0.213	0.334	0.276	0.149	0.193	0.267	0.220	0.150	-	-	-	-
✗	✓	✗	FZR$_{DistMult}$	0.234	0.346	0.293	0.174	0.208	0.281	0.239	0.163	-	-	-	-
✗	✓	✓	CZRL	0.244	0.370	0.314	0.164	0.199	0.280	0.233	0.151	-	-	-	-
✓	✓	✓	**SAZRL (ours)**	**0.270**	**0.400**	**0.343**	**0.196**	**0.217**	**0.296**	**0.251**	**0.172**	**0.152**	**0.336**	**0.233**	**0.071**
			Improvement (%)	10.66	6.67	9.24	5.95	4.33	5.34	5.02	5.52	360.61	401.49	762.96	446.15

Table 2: Details of existing ZRL datasets. $|\mathcal{A}|$ denotes the number of elements in set $\mathcal{A}$.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}_s\rvert$	$\lvert\mathcal{R}_{val}\rvert$	$\lvert\mathcal{R}_{test}\rvert$	$\lvert\mathcal{T}_s\rvert$	$\lvert\mathcal{T}_{val}\rvert$	$\lvert\mathcal{T}_{test}\rvert$
NELL-ZS	65,567	139	10	32	181053	1856	5483
Wiki-ZS	605,812	469	20	48	701977	7241	15710
FB15K-ZS	10399	191	23	21	209133	25174	17501

For evaluation, we follow the standard practice (Geng et al., 2021; Dong et al., 2024) and report Mean Reciprocal Rank (**MRR**) and **Hits@K** (K $\in 1, 5, 10$) to assess the performance of both baselines and our method.

5.2 Baselines and Implementation Details

We conduct extensive comparative experiments of SAZRL against state-of-the-art ZRL methods with publicly available code. The baselines include the ZRL variants of classical KGE models (**ZS-TransE** , **ZS-DistMult**, and **ZS-ComplEx**) from (Qin et al., 2020), text-based methods (**ZSGAN** (Qin et al., 2020), **CZRL** (Fang et al., 2025)), ontology-based methods (**OntoZSL** (Geng et al., 2021), **DOZSL** (Geng et al., 2022)) and expert-guided methods (**FZR** (Dong et al., 2024)). For DOZSL, we evaluate four variants combining ontology encoding strategies—neighborhood aggregation (AGG) or random initialization (RD)—with ZRL models using either GAN (Goodfellow et al., 2020) or GCN (Kipf and Welling, 2016). Following DOZSL, we also include **RGAT** (Busbridge et al., 2019), **DisenE** (Kou et al., 2020), and **DisenKGAT** (Wu et al., 2021)—using either GAN or GCN as the downstream ZRL models. For FZR (Dong et al., 2024), we evaluate two variants pretrained with TransE and DistMult, respectively.

More implementation details are provided in **Appendix** D.1, D.2, and D.3. The code and data are available at https://github.com/XKKuan/SAZRL.

5.3 Performance Comparison

The performance of SAZRL in comparison with ZRL baselines is presented in Table 1.

It is evident that SAZRL consistently outperforms all other methods across all benchmark datasets. Specifically, compared to the sub-optimal results, SAZRL yields an average relative improvement of 8.57% on NELL-ZS and 5.05% on Wiki-ZS across four metrics. Noting that these improvements are computed relative to the best overall results among all baselines, rather than those of individual models. Moreover, we categorize all methods based on the type of external knowledge they require: textual descriptions (**Text**), ontological schemas (**Onto**), and expert annotations (**Exp**). A key advantage of SAZRL is that it does not rely on them, making it particularly valuable in scenarios where KGs lack such resources. For instance, in FB15K-ZS where no external knowledge is available, SAZRL still achieves remarkable performance, while existing methods struggle to make predictions or even fail to run.

The effectiveness of SAZRL can be attributed to empirical regularities reported in (Lee et al., 2023), where semantically similar relations in a KG tend to share common entities, resulting in a natural clus-

tering effect. This provides a strong inductive bias that facilitates generalization from seen to unseen relations, eliminating the need for external semantic annotations. To further explore the question, "Why are structural patterns effective for ZRL?", we conduct an empirical analysis, which can be found in **Appendix** E. We also observe the similar regularities in our experiments (see **Appendix** F). Leveraging this, SAZRL constructs query graphs based on shared query entities, effectively capturing semantic correlations between seen and unseen relations. This enables SAZRL to infer meaningful representations for unseen relations by aggregating representations from their semantically related neighbors, thereby enabling robust ZRL reasoning.

5.4 Ablation Study

We conduct ablation studies to evaluate the contribution of each component in SAZRL on NELL-ZS and Wiki-ZS. The results are shown in Table 3. Specifically, for the **CQG** module, we examine the effect of two components: (1) **w/o query graph**: without the query graph, the representations of unseen relations are learnable and initialized using Glorot initialization. (2) **w/o subgraph**: only relations directly connected to the query entity are used to construct the query graph, without two-hop expansion. For the **ARU** module, we evaluate three variants: (3) **w/o similarity bias**: the global bias term $o^l_{I(k,j)}$ is removed from Eq. (6). (4) **w sum**: the representation of an unseen relation is computed by summing the representations of its neighboring seen relations. (5) **w avg**: the representation of an unseen relation is computed by averaging the representations of its neighbors.

Table 3: Results of the ablation experiments.

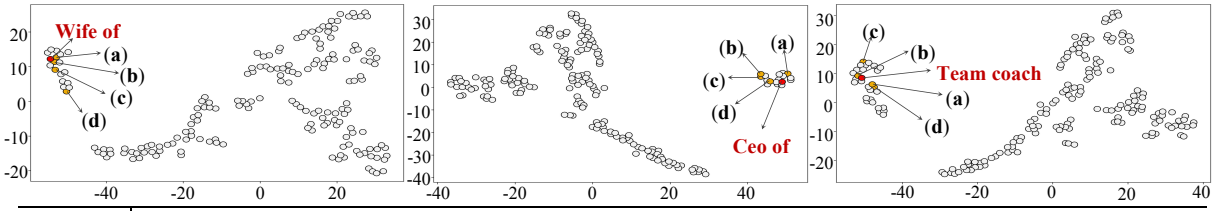
Data	Method	MRR	Hits@10	Hits@5	Hits@1
NELL-ZS	w/o query graph	0.018	0.029	0.019	0.009
	w/o subgraph	0.261	0.386	0.332	0.190
	w/o similarity bias	0.264	0.394	0.333	0.192
	w sum	0.246	0.36	0.307	0.183
	w avg	0.258	0.385	0.331	0.187
	SAZRL	**0.270**	**0.400**	**0.343**	**0.196**
Wiki-ZS	w/o query graph	0.035	0.061	0.043	0.016
	w/o subgraph	0.208	0.282	0.238	0.164
	w/o similarity bias	0.209	0.284	0.238	0.165
	w sum	0.208	0.283	0.237	0.165
	w avg	0.213	0.289	0.243	0.169
	SAZRL	**0.217**	**0.296**	**0.251**	**0.172**

Overall, each component of SAZRL contributes positively to its performance. Removing any component results in a performance drop to varying extents. Meanwhile, through the ablation experiments, we can draw the following conclusions: **(I)** From experiment (1), it is evident that the representations of unseen relations can be derived from seen ones. This process can be effectively achieved using only the structural patterns within the KG. **(II)** Experiment (2) shows that constructing a subgraph is beneficial. This may be because the query graph derived from the subgraph enables the model to better capture multi-hop neighboring relations of the unseen relations. Even if these relations are not directly connected to the unseen ones, there still exist latent indirect associations. The subgraph-based query graph facilitates the propagation of such associations during the model's learning process. **(III)** From experiment (3), we observe that the global bias positively contributes to the performance of SAZRL. This may be attributed to its role in helping the ARU capture global features of the query graph, improving the generalization across diverse query graphs. **(IV)** Experiments (4) and (5) indicate that the strength of the correlations between unseen and seen relations varies, which is consistent with real-world scenarios. Simply summing or averaging the representations is insufficient to capture the complex inter-relational dependencies between them.

5.5 Discussion on Structural Sparsity

Since SAZRL constructs the query graph based on shared entities between seen and unseen relations, we explore its robustness under structural sparsity (i.e., *unseen relations have few or no overlapping entities with seen ones*). Specifically, we vary the maximum number of shared entities per unseen relation in $\{0, 1, 5, 50\}$ and report the MRR on NELL-ZS and Wiki-ZS, as shown in Figure 4(a).

The results indicate that in the absence of shared entities, it becomes difficult to capture the correlations between seen and unseen relations, leading to poor performance. In contrast, the presence of shared entities leads to noticeable performance improvements, as they help establish meaningful connections between unseen and seen relations. An interesting observation is that the MRR does not change significantly as the number of shared entities increases from 1 to 50. One possible reason is that semantically similar relations exhibit a clustering effect: even when the shared entities differ, the neighbors tend to be similar. For example, in NELL-ZS, we compute the Jaccard and Overlap indexes of neighbors for the same query relation across different query entities. The average val-



Rank	1	2	3
Wife of	(a) Has spouse	(b) Mother of person; (c) Has family member	(d) Person born in location
Ceo of	(a) Person leads organization	(b) Agent collaborates with agent; (c) Mutual proxy for	(d) Agent controls
Team coach	(a) Organization has person; (b) Organization hired person	(c) Team member	(d) Worker

Figure 3: Results of case study. SAZRL effectively infers unseen relations by leveraging structural patterns in KGs.

ues are 0.39 and 0.72, respectively (refer to **Appendix** F for details). These results suggest that even with one shared entity, similar relations can still be captured to a certain extent.

To further validate this hypothesis, we conduct additional experiments under the condition that the number of shared entities is fixed at one. For each query relation, we randomly retain at most n $\in \{0, 1, 3, 10, 20\}$ neighbor relations and report the average MRR over five independent runs. The results are presented in Figure 4(b). A clear trend can be observed: as the number of neighbor relations increases, the performance of SAZRL gradually improves and gradually stabilizes. This is because when the number of neighbor relations is small (i.e., *neighbor sparsity*) (e.g., 1 or 3), the clustering effect weakens, the randomness and sparsity of neighbor relations increases, making it more difficult to establish reliable correlations and thus leading to degraded performance. In contrast, as the number of neighbor relations grows, the opposite effect occurs—the clustering effect strengthens, and randomness decreases, enabling SAZRL to effectively capture correlations. The results provide empirical support for the above hypothesis. Building on this observation, we also plan to explore strategies to mitigate neighbor sparsity in future.

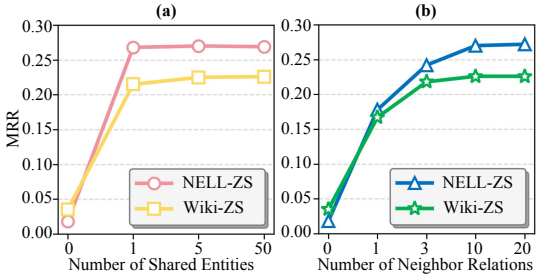


Figure 4: Discussion on Structural Sparsity.

5.6 Case Study

In this section, we present a case study to offer a more intuitive and concise qualitative analysis of SAZRL. Specifically, leveraging the semantic annotations provided by the NELL-ZS dataset, we report the top-3 seen relations most strongly associated with several unseen relations—namely, "Wife of", "CEO of", and "Team coach"—as learned by SAZRL. In addition, we visualize the clustering patterns of relation representations induced by SAZRL using t-SNE (Maaten and Hinton, 2008). The corresponding results are shown in Figure 3.

The results clearly show that: **(I)** SAZRL effectively infers unseen relations by leveraging structural patterns in the KG. For instance, given the unseen relation "Wife of," SAZRL retrieves semantically related seen relations from the background graph $\mathcal{T}_s$, such as "Has spouse", "Mother of person", "Has family member". Intuitively, these relations provide strong semantic cues that facilitate the inference of the target relation. This highlights the effectiveness of SAZRL in capturing semantic correlations between relations solely from structural information. **(II)** SAZRL has the ability to distinguish between relations. Across all three clustering results, it is evident that for a given unseen relation, its semantically related relations (either direct or indirect) are grouped together with it. In contrast, other relations are clearly separated and placed far from this cluster. **(III)** We further observe that the construction of the query graph in SAZRL is conditioned on different query relations, as described in section 4.1. As illustrated in Figure 3, distinct unseen relations exhibit noticeably different clustering patterns in the learned relation representations. This design exposes SAZRL to multiple relation-specific Query Graphs during training, thereby enhancing its generalization capability and mitigating the risk of overfitting to a single graph structure.

All of these observations provide strong empirical support for our initial motivation.

6 Conclusion and Future Work

In this paper, we propose SAZRL, a new paradigm that leverages structural patterns within KGs to facilitate ZRL. Extensive experimental results indicate that SAZRL consistently outperforms state-of-the-art ZRL methods while exhibiting strong potential for real-world applicability without relying on external knowledge. Inspired by this work, we plan to extend our research in the following directions: (1) exploring more robust relation representation strategies under neighbor sparsity, to better infer unseen relations when only limited structural context is available; (2) investigating more expressive structural encoding methods for capturing richer semantic correlations; and (3) extending ZRL to settings involving unseen entities, which presents a more challenging scenario for KGC.

7 Limitations

Since SAZRL performs inference solely based on the structural patterns of a given KG, it inevitably faces certain limitations. One limitation is data sparsity: semantically similar relation pairs may be missed if they do not share any entities, which prevents the model from capturing their correlations. Another limitation arises from unavoidable noise in KGs, as the construction process may introduce false-positive or false-negative facts, thereby adversely affecting structural inference.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (No.2025YJS045 and 2025JBZX064), the National Natural Science Foundation of China (Nos. U23B2062, and 82374302), and the State Key Laboratory on Technologies for Chinese Medicine Pharmaceutical Process Control and Intelligent Manufacture (SKL2024Z0102).

References

Shaked Brody, Uri Alon, and Eran Yahav. 2021. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. 2019. Relational graph attention networks. *arXiv preprint arXiv:1904.05811*.

Mingyang Chen, Wen Zhang, Yuxia Geng, Zezhong Xu, Jeff Z Pan, and Huajun Chen. 2023. Generalizing to unseen elements: A survey on knowledge extrapolation for knowledge graphs. *arXiv preprint arXiv:2302.01859*.

Zhuo Chen, Yichi Zhang, Yin Fang, Yuxia Geng, Lingbing Guo, Xiang Chen, Qian Li, Wen Zhang, Jiaoyan Chen, Yushan Zhu, and 1 others. 2024. Knowledge graphs meet multi-modal learning: A comprehensive survey. *arXiv preprint arXiv:2402.05391*.

Ziqiang Cui, Yunpeng Weng, Xing Tang, Fuyuan Lyu, Dugang Liu, Xiuqiang He, and Chen Ma. 2025. Comprehending knowledge graphs with large language models for recommender systems. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1229–1239.

Ling Ding, Lei Huang, Zhizhi Yu, Di Jin, and Dongxiao He. 2025. Towards global-topology relation graph for inductive knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11581–11589.

Zhijun Dong, Likang Wu, Kai Zhang, Ye Liu, Yanghai Zhang, Zhi Li, Hongke Zhao, and Enhong Chen. 2024. Fzr: Enhancing knowledge transfer via shared factors composition in zero-shot relational learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 497–507.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Zhiyi Fang, Hang Yu, Changhua Xu, Zhuofeng Li, Jie Ying, and Shaorong Xie. 2025. Contrastive zero-shot relational learning for knowledge graph completion. *Knowledge-Based Systems*, page 113425.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2024. Towards foundation models for knowledge graph reasoning. In *The Twelfth International Conference on Learning Representations*.

Xiou Ge, Yun Cheng Wang, Bin Wang, and C-C Jay Kuo. 2023. Compounding geometric operations for knowledge graph completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6947–6965.

Yuxia Geng, Jiaoyan Chen, Zhuo Chen, Jeff Z Pan, Zhiquan Ye, Zonggang Yuan, Yantao Jia, and Huajun Chen. 2021. Ontozsl: Ontology-enhanced zero-shot learning. In *Proceedings of the web conference 2021*, pages 3325–3336.

Yuxia Geng, Jiaoyan Chen, Wen Zhang, Yajing Xu, Zhuo Chen, Jeff Z. Pan, Yufeng Huang, Feiyu Xiong,

and Huajun Chen. 2022. Disentangled ontology embedding for zero-shot learning. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 443–453.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Xiaoyu Kou, Yankai Lin, Yuntao Li, Jiahao Xu, Peng Li, Jie Zhou, and Yan Zhang. 2020. Disene: Disentangling knowledge graph embeddings. *arXiv preprint arXiv:2010.14730*.

A Arun Kumar, Samarth Bhandary, Swathi Gopal Hegde, and Jhinuk Chatterjee. 2025. Knowledge graph applications and multi-relation learning for drug repurposing: A scoping review. *Computational Biology and Chemistry*, page 108364.

Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. 2023. Ingram: Inductive knowledge graph embedding via relation graphs. In *International Conference on Machine Learning*, pages 18796–18809. PMLR.

Mingchen Li, Chen Ling, Rui Zhang, and Liang Zhao. 2024. Zero-shot link prediction in knowledge graphs with large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, pages 753–760. IEEE.

Zhao Li, Xin Wang, Jun Zhao, Wenbin Guo, and Jianxin Li. 2025. Hycube: Efficient knowledge hypergraph 3d circular convolutional embedding. *IEEE Transactions on Knowledge and Data Engineering*.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, and 1 others. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckeschmidt. 2024. Anytime bottom-up rule learning for large-scale knowledge graph completion. *The VLDB Journal*, 33(1):131–161.

Pengda Qin, Xin Wang, Wenhu Chen, Chunyun Zhang, Weiran Xu, and William Yang Wang. 2020. Generative adversarial zero-shot relational learning for knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8673–8680.

Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–49.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pages 57–66.

Junkang Wu, Wentao Shi, Xuezhi Cao, Jiawei Chen, Wenqiang Lei, Fuzheng Zhang, Wei Wu, and Xiangnan He. 2021. Disenkgat: knowledge graph embedding with disentangled graph attention network. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 2140–2149.

Kuan Xu, Kuo Yang, Jian Liu, Xiangkui Lu, Jun Wu, and Xuezhong Zhou. 2025. Kgcrr: An effective metric-driven knowledge graph completion framework by designing a novel upper bound function with adaptive approximation to reciprocal rank. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12927–12935.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Jing Yang, Xiaowen Jiang, Yuan Gao, Laurence T Yang, and Jieming Yang. 2024. Generalize to fully unseen graphs: Learn transferable hyper-relation structures for inductive link prediction. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1274–1282.

Hang Yu and Haoyi Yu. 2025. Enhancing zero-shot knowledge graph relation prediction through large language models and contrastive learning. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 1490–1494.

A Differences between ZRL and fully inductive KGC.

A.1 Difference in Tasks

Although fully inductive KGC also involves reasoning over unseen relations, its objective is to evaluate a model's ability to generalize to unseen KGs, which is more appropriately characterized as a transfer learning problem across KGs (Galkin et al., 2024).

From the perspective of benchmark construction, in addition to the training, validation, and test splits, an extra **inference graph** is provided. Inductive KGC methods are trained on the training set and evaluated on the inference graph to assess their generalization ability across KGs. All entities and relations appearing in the validation and test sets are observed in the inference graph, which serves as the basis for evaluating a model's generalization performance (Lee et al., 2023).

In contrast to inductive KGC, ZRL exhibits a fundamentally different problem setting. ZRL benchmarks do not provide an inference graph, which implies that facts involving unseen relations appear only in the validation and test sets and are entirely unobservable for models. Consequently, the research question in ZRL is how to predict facts involving unseen relations when triples containing those relations are unavailable. From the perspective of handling unseen relations, ZRL is therefore better characterized as a problem of discovering and modeling unseen relations, rather than learning to generalize across KGs containing unseen relations, as in inductive KGC.

In short, inductive KGC assumes unseen relations are observable through an inference graph at inference time, whereas ZRL assumes no such observations are available. As a result, ZRL addresses the qualitatively different and more challenging problem of identifying and modeling entirely unseen relations without any relational instances. For a more detailed and formal comparison between these task settings, we refer readers to prior studies (Goodfellow et al., 2020; Lee et al., 2023; Galkin et al., 2024).

A.2 Difference in Methods

As discussed in the previous subsection, inductive KGC and ZRL differ fundamentally in task formulation and benchmark construction.

Our research on SAZRL is developed specifically for the ZRL setting. It is inspired by the intuition from inductive KGC that semantically related relations tend to share similar entity sets (Ding et al., 2025; Galkin et al., 2024; Lee et al., 2023). Importantly, this intuition is adopted at a conceptual level rather than by assuming access to relational instances or an inference graph. Accordingly, SAZRL introduces a task-specific graph construction and inference mechanism that is fundamentally different from those used in inductive KGC.

Due to the absence of an inference graph and the complete lack of relational instances for unseen relations in ZRL benchmarks, SAZRL cannot directly leverage structural information in the manner of inductive KGC methods. Consequently, despite the shared intuition, SAZRL differs fundamentally from the aforementioned inductive KGC approaches in the following aspects. Specifically, inductive KGC methods typically construct a single global relation graph over the entire inference graph. Such a design introduces several limitations when considered under the ZRL setting: **(1) it relies heavily on relational instances of unseen relations present in the inference graph; (2) the construction and maintenance of a global relation graph may not scale well to large-scale KGs; and (3) modeling all relations within a single relation graph increases the risk of overfitting.**

In contrast, SAZRL constructs query-specific graphs based on query pairs and observable entity anchors, without assuming access to an inference graph or relational instances of unseen relations. This design makes SAZRL naturally applicable to the ZRL setting and helps mitigate overfitting by avoiding reliance on a single global relation graph. Moreover, by operating on local subgraphs rather than the entire KG, SAZRL remains efficient and scalable on large-scale KGs.

B Algorithm

In this section, we present the detailed **training** and **testing** (Take, for example, predicting the tail entity t given a head–relation pair (h, r).) process of SAZRL in Algorithm 1 and Algorithm 2.To enhance clarity, some inherently parallel operations are illustrated using sequential loop constructs.

Algorithm 1 Training Process of SAZRL

Require: $\mathcal{T}_s$: training triplet set, num_{neg}: number of negative samples, θ: learnable parameters of SAZRL, lr: learning rate

while not converged **do**
  Shuffle the triplets in $\mathcal{T}_s$.
  for $\mathcal{T}_{tr}, \mathcal{T}_g \subset \mathcal{T}_s$ **do**
    $\mathcal{E}_q \leftarrow \emptyset$.
    for each $(e_i, r_k^{su}, e_j) \in \mathcal{T}_{tr}$ **do**
      Split (e_i, r_k^{su}, e_j) into (e_i, r_k^{su}) and (r_k^{su}, e_j). $\mathcal{E}_q^{r_k^{su}} \leftarrow \mathcal{E}_q^{r_k^{su}} \cup \{e_i, e_j\}$.
    end for
    $\mathcal{E}_{sub}^{r_k^{su}} \leftarrow \emptyset$.
    for each $e_i \in \mathcal{E}_q^{r_k^{su}}$ **do**
      Construct $\mathcal{E}_{2-hop}^{r_k^{su}}$ of e_i from $\mathcal{T}_g$.
      $\mathcal{E}_{sub}^{r_k^{su}} \leftarrow \mathcal{E}_{sub}^{r_k^{su}} \cup \mathcal{E}_{2-hop}^{r_k^{su}}$.
    end for
    Construct $\mathcal{S}_{sub}^{r_k^{su}}$ using Eq. 1.
    Construct $\mathcal{G}^{r_k^{su}}$ using Eq. 2-4.
    Initialize x_k^{su} and v_t for r_k^{su} and e_t in $\mathcal{G}^{r_k^{su}}$. $h_k^0 \leftarrow W' x_k^{su}$.
    for $l = 0, \ldots, L-1$ **do**
      Update h_k^{l+1} using Eq. 5.
    end for
    $z_k^{su} \leftarrow P h_k^L$. $\mathcal{L}_{mean} \leftarrow \emptyset$.
    for $(e_i, r_k^{su}, e_j) \in \mathcal{T}_{tr}$ **do**
      $\mathcal{S}_h \leftarrow \emptyset$, $\mathcal{S}_t \leftarrow \emptyset$.
      for $n = 1$ to num_{neg} **do**
        if replace head entity **then**
          $\mathcal{S}_h \leftarrow \mathcal{S}_h \cup \{(e_{i'}, r_k^{su}, e_j)\}$.
        end if
        if replace tail entity **then**
          $\mathcal{S}_t \leftarrow \mathcal{S}_t \cup \{(e_i, r_k^{su}, e_{j'})\}$.
        end if
      end for
      Define $\mathcal{L}$ according to Eq. 9.
      $\mathcal{L}_{mean} \leftarrow \mathcal{L}_{mean} \cup \{\mathcal{L}\}$.
    end for
    $\mathcal{L}_{mean} \leftarrow \text{mean}(\mathcal{L}_{mean})$.
    $\theta \leftarrow \text{Adagrad}(\nabla_\theta \mathcal{L}_{mean}, \theta, lr)$.
  end for
end while

Algorithm 2 Testing Process of SAZRL

Require: $\mathcal{T}_s$: Background triplet set, $\mathcal{T}_{test}$: Testing triplet set

while not end **do**
  for each query $(e_i, r_k^u, e_t) \in \mathcal{T}_{test}$ **do**
    $\mathcal{E}_q^{r_k^u} \leftarrow e_i$, $\mathcal{E}_{sub}^{r_k^u} \leftarrow \emptyset$.
    for each $e_i \in \mathcal{E}_q^{r_k^u}$ **do**
      Construct $\mathcal{E}_{2-hop}^{r_k^u}$ of e_i from $\mathcal{T}_s$.
      $\mathcal{E}_{sub}^{r_k^u} \leftarrow \mathcal{E}_{sub}^{r_k^u} \cup \mathcal{E}_{2-hop}^{r_k^u}$.
    end for
    Construct $\mathcal{S}_{sub}^{r_k^u}$ through Eq. 1.
    Construct $\mathcal{G}^{r_k^u}$ using $\mathcal{S}_{sub}^{r_k^u}$ and $(e_i, r_k^u, ?)$ through Eq. 2-4.
    Get x_k^u as representation for relation r_k^u.
    Set $h_k^0 \leftarrow W' x_k^u$.
    for $l = 0, \ldots, L-1$ **do**
      Get h_k^{l+1} using Eq. 5.
    end for
    $z_k^u \leftarrow P h_k^L$.
    Get score s_t of (e_i, r_k^u, e_t) through Eq. 8, $Scores \leftarrow \{s_t\}$
    for each candidate $e_j \in \mathcal{C}_(e_i, r_k^u)$ **do**
      Get s_j of (e_i, r_k^u, e_j) through Eq. 8.
      $Scores \leftarrow Scores \cup \{s_j\}$
    end for
    Get the rank of (e_i, r_k^u, e_t) from $Scores$.
  end for
end while

C Analysis on Different Conditions of The Constructed Query Graph

In this section, we investigate different configurations of the constructed query graphs in the CQG module. Specifically, in addition to the co-occurrence-based similarity (**CS**) among relations described in section 4.1, we evaluate two alternative measures: jaccard similarity (**JS**) and positive pointwise mutual information (**PPMI**) similarity.

For clarity, we follow the notation introduced in section 4.1. The JS score between relations r_k and $r_{k'}$ is then defined as:

$$s^{\text{JS}}(r_k, r_{k'}) = s_h^{\text{JS}}(r_k, r_{k'}) + s_t^{\text{Jac}}(r_k, r_{k'}), \quad (12)$$

$$s_h^{\text{JS}}(r_k, r_{k'}) = \frac{|H(r_k) \cap H(r_{k'})|}{|H(r_k) \cup H(r_{k'})|}, \quad (13)$$

$$s_t^{\text{JS}}(r_k, r_{k'}) = \frac{|T(r_k) \cap T(r_{k'})|}{|T(r_k) \cup T(r_{k'})|}, \quad (14)$$

$$H(r_k) = \{e_i \mid M_h[i,k] > 0\}, \quad (15)$$

$$T(r_k) = \{e_i \mid M_t[i,k] > 0\}, \quad (16)$$

where $M_h[i,k]$ and $M_t[i,k]$ denote the frequency matrices introduced in section 4.1, which record how often each entity appears as a head or tail entity, respectively. For PPMI, we first define joint and marginal probabilities for entities and relations at the head position as follows:

$$p_h(e_i, r_k) = \frac{M_h[i,k]}{\sum_{i',k'} M_h[i',k']}, \quad (17)$$

$$p_h(e_i) = \sum_k p_h(e_i, r_k), \quad (18)$$

$$p_h(r_k) = \sum_i p_h(e_i, r_k). \quad (19)$$

The PPMI is then computed as:

$$\text{PPMI}_h(e_i, r_k) = \max(\text{PMI}_h(e_i, r_k), 0), \quad (20)$$

$$\text{PMI}_h(e_i, r_k) = \log \frac{p_h(e_i, r_k)}{p_h(e_i) p_h(r_k)}. \quad (21)$$

Each relation r_k is the represented as a head-based PPMI-weighted vector:

$$\boldsymbol{\phi}_h(r_k) = [\text{PPMI}_h(e_1, r_k), ..., \text{PPMI}_h(e_n, r_k)]^T. \quad (22)$$

The head-based PPMI similarity between two relations is computed using cosine similarity:

$$s_h^{\text{PPMI}}(r_k, r_{k'}) = \frac{\boldsymbol{\phi}_h(r_k)^\top \boldsymbol{\phi}_h(r_{k'})}{\|\boldsymbol{\phi}_h(r_k)\|_2 \, \|\boldsymbol{\phi}_h(r_{k'})\|_2}. \quad (23)$$

The same procedure is applied to M_t to obtain $s_t^{\text{PPMI}}(r_k, r_{k'})$. The final PPMI similarity score is then defined as:

$$s^{\text{PPMI}}(r_k, r_{k'}) = s_h^{\text{PPMI}}(r_k, r_{k'}) + s_t^{\text{PPMI}}(r_k, r_{k'}). \quad (24)$$

For all similarity measures, the similarity scores are normalized to to sum to one.

We conduct comparative experiments on the NELL-ZS and Wiki-ZS datasets using these three similarity measures (CS, JS and PPMI), and report the performance of SAZRL in terms of MRR and Hits@10. The results are presented in Table 4.

Table 4: The performance of SAZRL with various similarity measures on NELL-ZS and Wiki-ZS.

Similarity	NELL-ZS		Wiki-ZS	
	MRR	Hits@10	MRR	Hits@10
CS	<u>0.270</u>	**0.400**	<u>0.217</u>	<u>0.296</u>
JS	0.266	0.394	0.213	0.291
PPMI	**0.272**	<u>0.399</u>	**0.218**	**0.299**

Overall, the results indicate that SAZRL is robust to the choice of relation similarity measure. All three similarity measures achieve comparable performance across both datasets, suggesting that the effectiveness of SAZRL primarily stems from its ability to exploit structural regularities in the knowledge graph rather than reliance on a specific similarity definition. Although PPMI slightly outperforms CS—consistent with the intuition that PPMI better suppresses the influence of high-frequency hub entities—the performance differences among similarity measures are minor. This may be partly because degree normalization already mitigates much of the frequency bias, and partly because the adaptive relation updating mechanism effectively compensates for imperfections in the initial similarity estimation. In contrast, Jaccard similarity consistently yields slightly lower performance, likely due to its sensitivity to sparse co-occurrence patterns.

D Implementation Details

D.1 Details For SAZRL

All experiments are conducted on a Linux server equipped with NVIDIA RTX 3090 GPUs (24GB RAM). For SAZRL, we adopt Adagrad (Duchi et al., 2011) as the optimizer, with the learning rate selected from {1e-4, 5e-4, 1e-3, 5e-3, 1e-2}. To ensure a sufficient search space, the embedding dimensions for both entities and relations are explored within $\{100, 200, 300, 400, 500\}$, and the hidden dimension of relations is set to twice the size of the initial relation embedding. For NELL-ZS and FB15K-ZS, the B is set to 5, and for Wiki-ZS, it is set to 10. During each training epoch, we randomly sample 500 triplets from $\mathcal{T}_s$ to construct $\mathcal{T}_{tr}$, with the remaining triplets forming the background KG $\mathcal{T}_g$. For negative sampling, we generate $1,000$ negative triplets per positive example on NELL-ZS and $3,000$ on Wiki-ZS and FB15K-ZS by randomly corrupting the head or tail entity. To ensure reliability, we conduct five independent experiments with the optimized hyperparameters and report the mean results.

D.2 Details For Baselines

For **ZS-TransE**, **ZS-DistMult**, and **ZS-ComplEx** proposed by (Qin et al., 2020), we report their results as presented in the paper (Qin et al., 2020), due to the lack of publicly available code and detailed experimental settings. For **ZSGAN** (Qin et al., 2020) and **OntoZSL** (Geng et al., 2021), we adopt DistMult (Yang et al., 2014) as the pre-training method to encode entity embeddings. We re-ran the experiments in our environment using the optimal hyperparameters provided in the released source code, to ensure fair and consistent evaluation. For **RGAT** (Busbridge et al., 2019), **DisenE** (Kou et al., 2020) and **DisenKGAT** (Wu et al., 2021), we conduct experiments based on the source code provided by (Geng et al., 2022). Along with **DOZSL** (Geng et al., 2022), we report the results conducted with the optimal hyperparameters provided in the source codes. For **FZR** (Dong et al., 2024), since it was not originally evaluated on the standard dataset splits, we reimplemented the experiments using the provided source code [3] on the standardized benchmarks. It is worth noting that we performed hyperparameter tuning for FZR to obtain the best possible results for a fair comparison. Specifically, following the standard FZR setup, we use *TransE* and *DistMult* as the pre-training models to encode the embeddings of entity pairs. We tune learning rates $\{lr_D, lr_D, lr_E\}$ over {5e-3, 1e-3, 5e-4, 1e-4, 5e-5}, respectively. In addition, we search the dimension of $fc1_{dim}$ within $\{200, 250, 300, 350\}$. Other hyperparameters are set to the optimal values reported in paper. For **CZRL**, since neither the official code nor the data split specifications are publicly available, we reimplemented the method based on the detailed computational descriptions provided in the original paper and its architectural similarity to **ZSGAN** (Qin et al., 2020). Experiments were conducted on datasets with standard splits. For shared hyperparameters, we followed the settings used in ZSGAN. In addition, we performed a grid search over the learning rate and λ_1, with learning rates selected from $\{1e{-}4, 5e{-}4, 1e{-}3, 5e{-}3, 1e{-}2\}$ and λ_1 from $\{0.2, 0.4, 0.6, 0.8\}$.

D.3 Baselines for FB15K-ZS

Since FB15K-ZS lacks external knowledge (e.g., textual descriptions, ontological schemas, or expert annotations), existing ZRL methods that rely on such knowledge cannot be applied. Therefore, when comparing to baselines, we evaluate their performance in the absence of external knowledge. Given that many ZRL methods share similar base models in the absence of external knowledge, we focus on comparing the performance of these base models. The models include: **TransE**, **DistMult**, **ComplEx**, **DisenE**, **RGAT**, and **DisenKGAT**. The corresponding results are shown at the respective ZRL variant positions.

[3] https://github.com/Junslow331/FZR

Table 5: Clustering effect of similar relations.

Query Realtion	Query Entity	Neighbor Relations	Jaccard Index	Overlap Index
Wife of	Female:Laura_bush	Has spouse, Has family member	0.503	0.873
Ceo of	Ceo:gregg_steinhafel	Worker, Person leads organization, Agent controls, Agent collaborates with agent, etc.	0.355	0.748
Team coach	Sportsteam:titans	Organization hired person, Team member, Organization has agent, Worker, etc.	0.445	0.815

E Why Structural Patterns Are Effective for ZRL?

In this work, we leverage the structural patterns inherent in KGs to infer unseen relations, using these patterns as a prior for ZRL. Below, we explain why structural patterns are effective for ZRL.

KGs encode human knowledge through structured relational interactions, rather than random connections. A key empirical observation is that semantically related relations tend to interact with similar sets of entities, leading to a clustering effect in the relational structure (Lee et al., 2023). This suggests that relation semantics are implicitly captured in entity co-occurrence patterns. Thus, the structure of a KG serves as an observable surrogate for latent relation semantics, where shared entity distributions across relations indicate high mutual information. Analyzing the relational structure significantly reduces uncertainty about the meaning of relations. SAZRL makes this implicit signal explicit by constructing query graphs. As a result, unseen relations can be inferred based on their proximity to structurally similar relations.

In summary, when a KG exhibits sufficient non-random organization, its structural regularities provide a strong inductive bias for generalizing from seen to unseen relations, eliminating the need for external semantic annotations.

F Clustering Effect of Similar Relations

In this section, we report the clustering effect of similar relations. Specially, semantically similar relations in a KG tend to share common entities (Lee et al., 2023), resulting in a clustering effect. Since the NELL-ZS dataset provides relation semantics, we conduct related experiments on it for ease of observation and analysis. Specifically, we take the query pairs **(Female:laura_bush, Wife of, ?)**, **(Ceo:gregg_steinhafel, Ceo of, ?)**, and **(Sportsteam:titans, Team coach, ?)** as examples, and examine the neighboring relations of their query relations in the background KG $\mathcal{T}_{tr}$, based on their query entities respectively. As shown in Table 5, the neighboring relations exhibit semantic similarity to the corresponding query relations. Moreover, for each of the three query relations, we compute the Jaccard and Overlap indices of neighboring relations by considering all possible pairs of query entities in $\mathcal{T}_{\text{test}}$, and report their average values. The Jaccard index and Overlap index can be defined as: $\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}$, $\text{Overlap}(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)}$, where A and B denote two sets of neighboring relations. The results indicate that, despite variations in query entities, majority of neighboring relations remain semantically similar to the query relation. We further extend this analysis to all query relations in $\mathcal{T}_{test}$, and present the distributions of the Jaccard and Overlap indices. As illustrated in the Figure 5, the neighboring relation sets for most query relations exhibit high similarity, by considering all possible pairs of query entities. Although some cases show relatively low Jaccard indices, their Overlap indices remain high. This phenomenon is likely due to KG incompleteness, which results in neighbor sparsity as discussed in section 5.5.

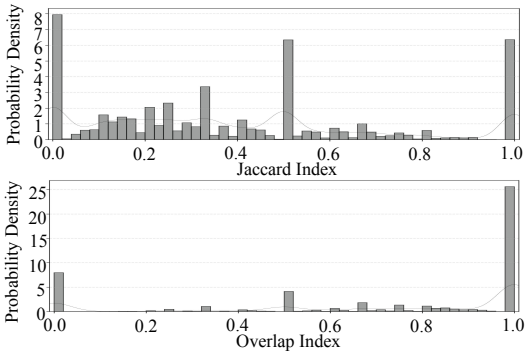


Figure 5: Clustering effect of similar relations.

Collectively, these findings suggest that the clustering behavior of semantically similar relations is preserved across different entities. This observation aligns with the phenomenon reported in (Lee et al., 2023), where semantically related relations in a KG tend to share common entities.