

# DUAL DENOISING LOGICAL REASONING FOR INDUCTIVE KNOWLEDGE GRAPH COMPLETION

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

In recent years, message-passing Graph Neural Networks (GNNs) have been widely used to tackle the problem of inductive knowledge graph completion. Though great progress has been made in GNN-based knowledge graph reasoning, it still suffers from the noise existing in irrelevant entities. These noises accumulated exponentially as the reasoning process continues, significantly impacting the overall performance of the model. To tackle this problem, several node-based sampling methods have been proposed for denoising. However, they do have inherent limitations. Firstly, they rely on node scores to evaluate node importance, which cannot effectively assess the quality of paths in GNN-based reasoning. Secondly, they often overlook noise interference caused by irrelevant edges. To address these problems, we propose a dual denoising logical reasoning (DDLRL) framework, which integrates path-based and edge-based sampling to achieve comprehensive denoising. Specifically, DDLRL employs a path-scoring mechanism to evaluate the importance of paths, aiming to remove irrelevant paths. Moreover, DDLRL leverages rules within the knowledge graph to remove irrelevant edges. Through the dual denoising process, we can achieve more effective logical reasoning. To demonstrate the effectiveness of the DDLRL framework, we conduct experiments on three benchmark datasets, and our approach achieves state-of-the-art performance.

## 1 INTRODUCTION

Knowledge Graphs (KGs) are widely used to formulate the structure information among real-world entities. They employ nodes to represent entities and edges to capture their relations. Despite their utility, KGs are often afflicted by inherent incompleteness and inaccuracies. To address these limitations, the vital task of Knowledge Graph Completion (KGC) has emerged. KGC is dedicated to inferring missing or latent facts within KGs and has been extensively explored in prior research (Bordes et al., 2013; Galárraga et al., 2015; Ho et al., 2018). This task holds particular importance in various applications, such as question answering (Zhang et al., 2018), natural language processing (Zhang et al., 2019), and recommendation systems (Wang et al., 2018), where the precision and comprehensiveness of knowledge representation are paramount for achieving high-quality results.

Various methods have been developed for KGC to predict missing facts, including TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), and ConvE (Dettmers et al., 2018). These methods map entities and relations to low-dimensional spaces and predict missing relations through tensor operations. However, traditional KGC models are inherently transductive, requiring the presence of all entities in the training data. Since knowledge graphs evolve over time, inductive knowledge graph completion becomes a hot topic, where new entities emerge. In such cases, traditional embedding-based models struggle to adequately represent emerging entities and perform poorly in inductive KGC.

Inductive KGC is a challenging task because new entities emerge in the testing phase, which is not seen during the training process. Therefore, how to effectively represent the new entities becomes a problem. Some works utilize GNNs to acquire meaningful representations of entities within knowledge graphs. (Teru et al., 2020) proposed GraIL, aiming to implicitly derive logical rules by analyzing the subgraph surrounding a candidate triplet in an entity-independent manner. Subsequent research efforts have enhanced this process by considering topological relations (Chen et al., 2021)

or optimizing message passing through node-edge interactions (Mai et al., 2021). NBFNet (Zhu et al., 2021) and RED-GNN (Zhang & Yao, 2022) introduce a path-based approach for direct learning of relation representations, eliminating the need to learn entity embeddings. Despite the good performance of these methods, they follow the same GNN message-passing process, i.e., propagating messages freely through all edges in graphs, which would bring in noise. In order to minimize the influence of noise, (Zhang et al., 2023a) propose to select the top K nodes in each iteration using learnable attention mechanisms. Nonetheless, there are two limitations to the existing studies. Firstly, take the knowledge graph in Fig 1 as an example, we can easily deduce that the answer to the query  $(a, \text{Mother}, ?)$  is  $d$  because there exists an evidence path  $Father(a, b) \wedge Wife(b, d)$ . However, according to the top-K strategy, the result obtained is  $e$ , because node  $e$  has a higher score than node  $d$ . Moreover, this strategy might also deduce the correct answer through pseudo-evidence reasoning, as exemplified by the path  $Friend(a, f) \wedge Mother(f, g) \wedge Friend(g, d)$ . Secondly, from Fig 1, we can also intuitively observe that the relevance of various relations within the red path to the target relation is higher than the relevance of various relations within the green path to the target relation. Therefore, the correlation between relations within the path and the target relation helps the model filter out irrelevant reasoning paths (i.e., the green path). It is evident that the current methods have not taken this into account.

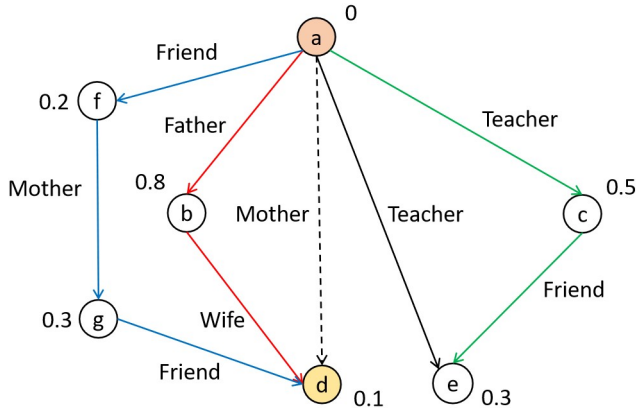


Figure 1: An example of KG reasoning, where the query is  $(a, \text{Mother}, ?)$  and the answer is  $d$ . Green/Blue lines represent irrelevant paths (i.e., non-evidence path) to this query, while the red line is relevant (i.e., evidence path). The dashed line indicates a missing fact, and numerals represent node scores.

To address these challenges, this paper proposes a **Dual Denoising Logical Reasoning (DDLR)** framework, which achieves comprehensive denoising through path and edge sampling. Specifically, for path sampling, we designed a novel path-scoring mechanism, which takes into account not only the current path score but also the scores of the remaining paths, to evaluate the importance of the path for sampling paths most relevant to the objective. For edge sampling, we leverage single rules (i.e., ordered relation pairs), which reveal correlations between relations, to explore the most relevant edges while filtering out irrelevant paths. In addition, to avoid the information loss caused by the top-k strategy, we adopt a sampling method that combines the Bernoulli distribution and rule confidence to sample edges. In our experiments, we observed that this sampling strategy can preserve promising targets during the reasoning process, thus enhancing the model’s performance.

To summarize, the main contributions of this paper are as follows:

- We propose a dual denoising logical reasoning model that comprehensively addresses noise interference in inductive KGC by both path sampling and edge sampling.
- We propose a path-scoring mechanism to assess the effectiveness of paths, enabling the precise selection of paths relevant to the target while simultaneously discarding irrelevant ones. This innovation enhances the reasoning capability of DDLR within knowledge graphs.
- We extract single rules at the triplet level and apply them to the edge sampling process. In contrast to conventional methods, it places a stronger emphasis on finer-grained triplets, thereby improving rule credibility and making sampling more logical.

- We conduct extensive experiments on three benchmark datasets to assess the effectiveness of DDLR. The results illustrate that our proposed algorithm surpasses state-of-the-art baselines in terms of prediction accuracy.

## 2 RELATED WORK

### 2.1 GNNs ON KNOWLEDGE GRAPHS

Significant progress (Yang et al., 2015; Schlichtkrull et al., 2018; Toutanova et al., 2015b; Dettmers et al., 2018; Socher et al., 2013; Wang et al., 2019; García-Durán et al., 2017) has been made in addressing the challenge of multiple relationship edges in knowledge graph completion through the extension of GNNs. These efforts have led to innovative message-passing operations that are carefully designed to accurately capture complex relationship information during feature transformation and aggregation processes. Prominent knowledge graph completion models include RGCN (Schlichtkrull et al., 2018), which enhances reasoning capabilities by utilizing relation-specific transformation matrices; CompGCN (Vashishth et al., 2020), which seamlessly integrates neighborhood information through entity-relation composition operations; KBGAT (Yang et al., 2015), a groundbreaking model that uses attention mechanisms to differentiate entity roles in multiple relationships; and recently introduced models in this field such as NBFNet (Zhu et al., 2021) and RED-GNN (Zhang & Yao, 2022), which introduce progressive propagation strategies from query entities to their multi-hop neighbors. These models have shown potential in both inductive and transductive reasoning scenarios and have demonstrated efficiency in evaluating multiple candidate entities. Additionally, RuleNet (Zhang et al., 2023b) has achieved remarkable results by effectively capturing potential relationship connections. However, the number of irrelevant entities involved exponentially increases with deeper propagation paths, making it more difficult to identify target answer entities.

### 2.2 SAMPLING METHODS FOR GNNs

Traditional message propagation methods face scalability issues when dealing with large-scale Graph Neural Networks (GNNs) (Hu et al., 2020). To address this challenge, several sampling methods have been proposed. For instance, GraphSAGE (Hamilton et al., 2017) and PASS (Chen et al., 2018a) sample a fixed-size neighborhood set from the complete neighbors of each node at each propagation step, while FastGCN (Yang et al., 2020), Adaptive-GCN (Chen et al., 2018b), and LADIES (Shi et al., 2019) sample an equal number of nodes at each propagation step. On the other hand, Cluster-GCN (Chiang et al., 2019), GraphSAINT (Hamilton et al., 2020), and ShadowGNN (Ma et al., 2021) extract subgraphs containing highly relevant nodes around anchor nodes. To address the challenge of convergence speed, a new subgraph sampling method with guaranteed convergence has recently been proposed (Shi et al., 2023). In contrast, the LAYER-neighBOR sampling (LABOR) (Balın & Ümit V. Çatalyürek, 2022) algorithm aims to reduce the number of sampled vertices by directly replacing neighborhood sampling with the same diffusion hyperparameters while maintaining quality (Balın & Ümit V. Çatalyürek, 2022). Adaprop (Zhang et al., 2023a) designs an incremental sampling mechanism to preserve proximate and promising targets. However, they overlook the impact of edges on sampling. In contrast, our proposed sampling strategy comprehensively considers the influence of both paths and edges on sampling, thereby improving the model’s performance.

## 3 PRELIMINARY

**Knowledge Graph Completion.** A knowledge graph, denoted as  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ , comprises finite sets of entities (nodes) represented as  $\mathcal{V}$ , facts (edges) as  $\mathcal{E}$ , and relation types as  $\mathcal{R}$ . Each fact is a triplet  $(x, r, y) \in \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ , signifying the presence of relation  $r$  from entity  $x$  to entity  $y$ . Knowledge graph completion is a task focused on predicting answers for queries such as  $(u, q, ?)$  or equivalently,  $(u, q^{-1}, ?)$ , where  $q^{-1}$  represents the inverse relation of  $q$ . Given a query  $(u, q, ?)$ , the objective is to determine the set of answers  $\mathcal{V}(u, q, ?)$  such that for all  $v \in \mathcal{V}(u, q, ?)$ , the triplet  $(u, q, v)$  holds true.

**Path-based Approaches.** Path-based approaches achieve knowledge graph completion by analyzing the paths between a pair of entities within a knowledge graph. For example, a path

$Father(a, b) \wedge Wife(b, d)$  may be used to predict  $Mother(a, d)$  in Fig 1. From a representation learning perspective, path-based approaches aim to learn a representation  $\mathbf{h}_q(u, v)$  to predict the triplet  $(u, q, v)$  based on all paths  $\mathcal{P}_{u \rightarrow v}$  from entity  $u$  to entity  $v$ .

$$\mathbf{h}_q(u, v) = \bigoplus_{P \in \mathcal{P}_{u \rightarrow v}} \mathbf{h}_q(P) = \bigoplus_{P \in \mathcal{P}_{u \rightarrow v}} \bigotimes_{(x, r, y) \in P} \mathbf{w}_q(x, r, y) \quad (1)$$

where  $\bigoplus$  is a permutation-invariant aggregation function,  $\bigotimes$  is an aggregation function.  $\mathbf{w}_q(x, r, y)$  is the representation of triplet  $(x, r, y)$  conditioned on the query relation  $q$ .

## 4 METHOD

In this section, we provide a detailed introduction to the DDLR framework, which employs dual denoising for more efficient logical reasoning. The framework, as shown in Fig 2, consists of several iterative sampling units. Each iterative sampling unit comprises two sub-modules: path-based sampling and edge-based sampling. DDLR employs these iterative sampling units to remain the most relevant paths or edges for the task.

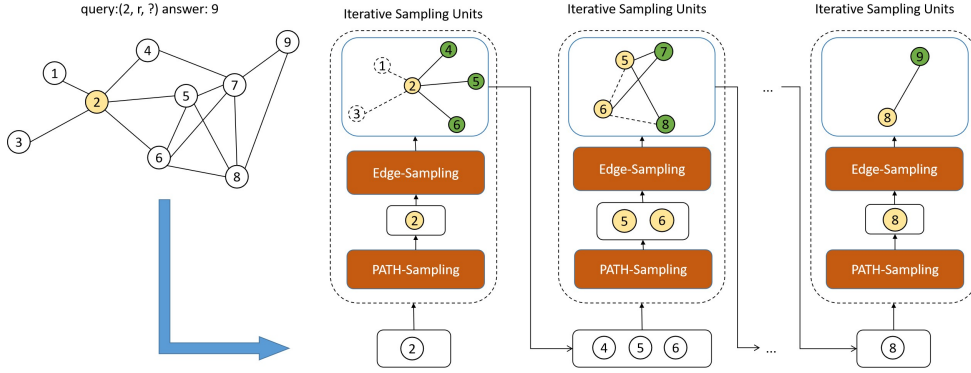


Figure 2: An overview of DDLR. For simplicity, we use circles to represent entities and ignore the types of relation. Dashed lines and dashed circles represent unrelated relations and entities, respectively, yellow circles represent the ending nodes of important paths, while the green circles represent nodes awaiting sampling. DDLR improves the model’s performance by selecting the most relevant paths and edges through path-based and edge-based sampling at each step.

### 4.1 PATH SAMPLING

**Important Paths for Reasoning.** Given a query relation and a lot of entities, only some of the paths between the entities are important for answering the query. Take Fig 1 as an example, it is clear that only evidence path  $Father(a, b) \wedge Wife(b, d)$  can lead to the deduction that triplet  $Mother(a, d)$  holds true. Formally, we define  $\mathcal{P}_{u \rightarrow v|q} \subseteq \mathcal{P}_{u \rightarrow v}$  to be the set of paths from  $u$  to  $v$  that is important to the query relation  $q$ .

$$\mathbf{h}_q(u, v) = \bigoplus_{P \in \mathcal{P}_{u \rightarrow v}} \mathbf{h}_q(P) \approx \bigoplus_{P \in \mathcal{P}_{u \rightarrow v|q}} \mathbf{h}_q(P) \quad (2)$$

In other words, any path  $P \in \mathcal{P}_{u \rightarrow v} \setminus \mathcal{P}_{u \rightarrow v|q}$  with respect to  $\mathbf{h}_q(u, v)$  is considered as noise, meaning it is not important to obtain the right answer. Therefore, if we calculate the representation  $\mathbf{h}_q(u, v)$  using only the important paths  $\mathcal{P}_{u \rightarrow v|q}$ , we can enhance the model’s performance.

**Path Scoring.** Given a query of the form  $(u, q, ?)$ , our aim is to identify the important paths  $\mathcal{P}_{u \rightarrow v|q}$  for all entities  $v$  in the knowledge graph. Nevertheless, extracting these important paths from the extensive set  $\mathcal{P}_{u \rightarrow v}$  is highly challenging due to its high complexity. So we employ an iterative node selection process to identify the paths that cover the important paths.

$$\hat{\mathcal{P}}_{u \rightarrow v|q}^{(0)} = \{(u, \text{selfloop}, v)\} \quad \text{if } u = v \text{ else } \emptyset \quad (3)$$

$$\hat{\mathcal{P}}_{u \rightarrow v|q}^{(t)} = \bigcup_{x \in n_{uq}^{(t-1)}(\hat{\mathcal{P}}_{u \rightarrow x|q}^{(t-1)}), (x, r, v) \in \mathcal{N}(x)} \{P + \{(x, r, v)\} | P \in \hat{\mathcal{P}}_{u \rightarrow x|q}^{(t-1)}\} \quad (4)$$

where  $\hat{\mathcal{P}}_{u \rightarrow v|q}^{(t)}$  computed by the above iteration is a superset of the important paths  $\mathcal{P}_{u \rightarrow v|q}^{(t)}$  of length  $t$ ,  $n_{uq}^{(t-1)}(\hat{\mathcal{P}}_{u \rightarrow x|q}^{(t-1)})$  selects ending nodes of important paths  $\mathcal{P}_{u \rightarrow x|q}^{(t-1)}$ .  $\mathcal{N}(x)$  is the neighborhood of node  $x$ .

Next, we learn a representation  $\mathbf{h}_q^{(t)}(u, v)$  based on the paths  $\hat{\mathcal{P}}_{u \rightarrow v|q}^{(t)}$  of length  $t$ .

$$\mathbf{h}_q^{(t)}(u, v) = \bigoplus_{P \in \hat{\mathcal{P}}_{u \rightarrow v|q}^{(t)}} \mathbf{h}_q^{(t)}(P) = \bigoplus_{P \in \hat{\mathcal{P}}_{u \rightarrow v|q}^{(t)}} \bigotimes_{(x, r, y) \in P} \mathbf{w}_q(x, r, y) \quad (5)$$

Then, we utilize a learnable path scoring function  $s_{uq}^{(t)}$  to evaluate paths of length  $t$ . However, evaluating paths from the extensive set  $\hat{\mathcal{P}}_{u \rightarrow v|q}^{(t)}$  presents a significant challenge due to its exponential size. Therefore, we approximate the evaluation of paths by evaluating the nodes.

$$s_{uq}^{(t)}(\hat{\mathcal{P}}_{u \rightarrow v|q}^{(t)}) \approx s_{uq}^{(t)}(\hat{\mathcal{V}}^{(t)}) \quad (6)$$

where  $\hat{\mathcal{V}}^{(t)}$  denotes the end nodes of the  $\hat{\mathcal{P}}_{u \rightarrow v|q}^{(t)}$ .

$s_{uq}^{(t)}(x)$  is jointly determined by the current path score  $d_q^{(t)}(u, x)$ , representing the path score from node  $u$  to node  $x$ , and the remaining path score  $r_q^{(t)}(x, v)$ , representing the path score from node  $x$  to node  $v$ . Although it is feasible to derive  $r_q^{(t)}(x, v)$  from the representation of the remaining path, obtaining the representation is extremely difficult because we do not possess the representation of the answer entity  $v$ . Hence, we utilize the query relation  $q$  as an approximation to get the representation of the remaining path.

$$\mathbf{r}_q^{(t)}(x, v) = \mathbf{h}_q^{(t)}(u, x) \otimes g([\mathbf{h}_q^{(t)}(u, x), \mathbf{q}]) \quad (7)$$

where  $\mathbf{r}_q^{(t)}(x, v)$  denotes the representation of the remaining path,  $g(\cdot)$  is a feed-forward network and  $[\cdot, \cdot]$  concatenates two representations,  $\otimes$  denotes element-wise multiplication.

To achieve a more precise evaluation of paths, it is necessary to take into account the scores of previous nodes within the path.

$$s_{uq}^{(0)}(u) = 0 \quad (8)$$

$$s_{uq}^{(t)}(x) = r_q^{(t)}(x, v) \oplus d_q^{(t)}(u, x) = \sigma(f(\mathbf{r}_q^{(t)}(x, v))) \oplus \bigoplus_{(\hat{x}, r, x) \in \mathcal{N}(x)} s_{uq}^{(t-1)}(\hat{x}) \quad (9)$$

where  $\sigma$  is the sigmoid function,  $f(\cdot)$  is a feed-forward network.  $d_q^{(t)}(u, x)$  incorporates the scores of previous nodes within the path.

Finally, we simply pick the top-K paths (i.e., nodes) according to the current scoring function  $s_{uq}^{(t)}(x)$ .

$$n_{uq}^{(t)}(\hat{\mathcal{P}}_{u \rightarrow x|q}^{(t)}) = n_{uq}^{(t)}(\hat{\mathcal{V}}^{(t)}) = \arg \max_{N' \subseteq \hat{\mathcal{V}}^{(t)}, |N'|=K} \sum_{x \in N'} s_{uq}^{(t)}(x) \quad (10)$$

In order to perform reasoning, we need to compute the representation  $\mathbf{h}_q^{(t)}(u, v)$  based on the important paths. Then Eq 5 is modified as follows:

$$\mathbf{h}_q^{(0)}(u, v) = \mathbf{q} \quad \text{if } u = v \text{ else } \mathbf{0} \quad (11)$$

$$\mathbf{h}_q^{(t)}(u, v) = \bigoplus_{x \in n_{uq}^{(t-1)}(\hat{\mathcal{V}}^{(t-1)}), (x, r, v) \in \mathcal{N}(x)} \mathbf{h}_q^{(t-1)}(u, x) \otimes \mathbf{w}_q(x, r, v) \quad (12)$$

where  $\mathbf{q}$  denotes the representation of relation  $q$ .

## 4.2 EDGE SAMPLING

**Bernoulli Sampling Based On Single Rules.** In addition to investigating the impact of important paths for reasoning, we also explore the influence of relations (i.e., edges) in the reasoning process. For instance, as depicted in Fig. 1, we can observe that the relations within evidence paths exhibit a higher degree of relevance to the target relation compared to those in non-evidence paths. This observation motivates us to sample relations with higher relevance, aiming to assist the model in selecting the most pertinent paths during reasoning. We consider a direct approach to sample important relations by randomly selecting relations from the graph. Formally, we sample a modified subset  $\tilde{\mathcal{R}}^{(t)}$  from the candidate relations  $\hat{\mathcal{R}}^{(t)}$  with certain probabilities at the t-hop.

$$\tilde{\mathcal{R}}^{(t)} = \{r \mid r \in \hat{\mathcal{R}}^{(t)}, \text{Bern}(p_{rq}^{(t)}) = 1\} \quad (13)$$

where  $p_{rq}^{(t)}$  is the probability of relevance between  $r$  and  $q$ , *Bern* represents a Bernoulli distribution,  $\hat{\mathcal{R}}^{(t)}$  denotes the set of candidate relations, take Fig 1 as an example,  $\{\text{Mother}, \text{Wife}, \text{Friend}\}$  is the candidate relation set for the model at the second hop.  $\tilde{\mathcal{R}}^{(t)}$  is then used as the relation set in the generated view.  $p_{rq}^{(t)}$  should reflect the importance of the relation  $r$ , depending on which the model can discard irrelevant relations while retaining those most relevant to the task.

Following logical notations, we denote potential relevance between relation  $r_1$  and relation  $r_2$  using a single rule  $r_1 \Rightarrow r_2$ . To quantify the degree of such relevance, we define the confidence of a single rule as follows:

$$\mathcal{C}(r_1 \Rightarrow r_2) = \frac{\sum_{t \in \mathcal{E}} \mathbb{1}(r_1 \in \mathbf{E}_r(t) \wedge r_2 \in \mathbf{E}_r(t))}{\sum_{t \in \mathcal{E}} \mathbb{1}(r_1 \in \mathbf{E}_r(t))} \quad (14)$$

where the function  $\mathbb{1}(x)$  equals 1 when  $x$  is true and 0 otherwise,  $\mathbf{E}_r$  extracts relations from the triplets. As an empirical statistic over the entire KG,  $\mathcal{C}(r_1 \Rightarrow r_2)$  is larger if more triplets with relation  $r_1$  also have  $r_2$ .

Next, we calculate the probability of each relation based on its confidence value. The probabilities can then be obtained after a normalization step that transforms the values into possibilities, which is defined as

$$p_{rq}^{(t)} = \min\left(\frac{\mathcal{C}_{max}^{(t)} - \mathcal{C}(r \Rightarrow q)}{\mathcal{C}_{max}^{(t)} - \mathcal{C}_{avg}^{(t)}} \cdot p_e, p_\tau\right) \quad (15)$$

where  $p_e$  is a hyperparameter that controls the overall probability of important relations,  $\mathcal{C}_{max}^{(t)}$  and  $\mathcal{C}_{avg}^{(t)}$  represent the maximum and average values, respectively, within the confidence set  $\mathcal{C}^{(t)}$  associated with  $\tilde{\mathcal{R}}^{(t)}$ . and  $p_\tau$  is a truncation probability used to cut off probability values because extremely low probabilities can result in the loss of important relations.

### 4.3 COMBINATION

Path-based sampling allows the model to select the important paths for reasoning, while edge-based sampling selects important edges. By employing both types of sampling, the model achieves effective denoising for knowledge graph completion. Then, we modify Eq 12 as follows:

$$\mathbf{h}_q^{(t)}(u, v) = \bigoplus_{x \in \mathcal{n}_{uq}^{(t-1)}(\hat{\mathcal{V}}^{(t-1)}), r \in \tilde{\mathcal{R}}^{(t-1)}, (x, r, v) \in \mathcal{N}(x)} \mathbf{h}_q^{(t-1)}(u, x) \otimes \mathbf{w}_q(x, r, v) \quad (16)$$

Eq. 16 encourages the model to select important paths for reasoning. A pseudo-code of the algorithm is illustrated in the appendix.

### 4.4 LOSS FUNCTION

We train the model to score positive triples higher than the negative triples using a multi-class log-loss following Lacroix et al. (2018). The loss function  $\mathcal{L}$  is

$$f(u, r, v) = \mathbf{w}_s^T \mathbf{h}_q^{(L)}(u, v) \quad (17)$$

$$\mathcal{L} = \sum_{(u, r, v) \in \mathcal{E}_{train}} \left( -f(u, r, v) + \log \left( \sum_{\forall x \in \mathcal{E}} \exp(f(u, r, x)) \right) \right), \quad (18)$$

where  $\mathbf{w}_s \in \mathbb{R}^d$  is a weight parameter,  $\mathbf{h}_q^{(L)}(u, v)$  represents the embedding of entity  $v$  at step  $L$ ,  $\mathcal{E}_{train}$  denotes the set of the positive triples  $(u, r, v)$ .

## 5 EXPERIMENTS

### 5.1 EXPERIMENT SETUP

**Datasets.** We conducted experiments on three widely recognized datasets: WN18RR (Dettmers et al., 2018), FB15k-237 (Toutanova et al., 2015a), and Nell-995 (Xiong et al., 2017), which are commonly employed for transductive relation prediction tasks. To assess inductive relation prediction, we followed the methodology of Teru et al. (Teru et al., 2020), who created four variations of each dataset. Each inductive dataset consists of a distinct train set and test set, ensuring that their entity compositions do not overlap. For further details, please refer to the supplement.

**Baselines.** In the inductive setting, methods that involve learning entity embeddings during training are not applicable. Consequently, we conduct a comparative analysis with path-based approaches that focus on acquiring sequential rules, including RuleN (Meilicke et al., 2018), NeuralLP (Yang et al., 2017), and DRUM (Sadeghian et al., 2019). Furthermore, we include GNN-based methods such as GraIL (Teru et al., 2020), NBFNet (Zhu et al., 2021), RED-GNN (Zhang & Yao, 2022), Adaprop (Zhang et al., 2023a), and GraPE (Wang et al., 2023) as our baseline models.

**Details.** For the DDLR model, we perform hyperparameter tuning on a variety of settings, including the learning rate within the range of  $[10^{-4}, 10^{-2}]$ , weight decay within the range of  $[10^{-5}, 10^{-2}]$ , dropout rate within the range of  $[0, 0.3]$ , batch size chosen from  $\{5, 10, 20, 50, 100\}$ , dimension  $d$  chosen from  $\{32, 48, 64, 96\}$ ,  $d_{att}$  for attention chosen from  $\{3, 5\}$ , layer  $L$  chosen from  $\{3, 4, 5, 6, 7\}$ , the number of sampled entities  $K$  among  $\{50, 100, 150, 200, 250, 300\}$ , activation function  $\delta$  chosen from  $\{identity, tanh, ReLU\}$ , probability multiplier  $p_e$  chosen from  $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$  and cut-off probability  $p_\tau$  chosen from  $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ . We adopt Adam (Kingma & Ba, 2014) as the optimizer. Please refer to the supplement for more detailed information.

**Results.** The performance comparison presented in Table 1 underscores that, despite its inductive reasoning capabilities, GraIL falls notably short of RED-GNN, NBFNet, Adaprop, and GraPE. The patterns acquired by GraIL exhibit limited generalization to unseen knowledge graphs. In contrast,

Table 1: The performance of inductive reasoning on three datasets, where the best results are in **bold** and the second best results are underlined.

metrics	methods	WN18RR				FB15k237				NELL-995			
		V1	V2	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4
MRR(%)	RuleN	66.8	64.5	36.8	62.4	36.3	43.3	43.9	42.9	61.5	38.5	38.1	33.3
	Neural LP	64.9	63.5	36.1	62.8	32.5	38.9	40.0	39.6	61.0	36.1	36.7	26.1
	DRUM	66.6	64.6	38.0	62.7	33.3	39.5	40.2	41.0	62.8	36.5	37.5	27.3
	GraIL	62.7	62.5	32.3	55.3	27.9	27.6	25.1	22.7	48.1	29.7	32.2	26.2
	NBFNet	68.4	65.2	42.5	60.4	30.7	36.9	33.1	30.5	58.4	41.0	42.5	28.7
	RED-GNN	70.1	69.0	42.7	65.1	36.9	46.9	44.5	44.2	63.7	41.9	43.6	36.3
	AdaProp	73.3	71.5	47.4	66.2	31.0	47.1	47.1	45.4	64.4	45.2	43.5	36.6
	GraPE	74.2	70.7	47.2	65.3	41.5	48.8	48.1	<b>47.0</b>	77.7	<b>49.4</b>	45.0	38.3
	DDLRL	<b>79.4</b>	<b>79.8</b>	<b>52.0</b>	<b>76.4</b>	<b>48.8</b>	<b>49.7</b>	<b>48.8</b>	46.3	<b>81.1</b>	48.2	<b>46.5</b>	<b>40.6</b>
	RuleN	63.5	61.1	34.7	59.2	30.9	34.7	34.5	33.8	54.5	30.4	30.3	24.8
Hit@1 (%)	Neural LP	59.2	57.5	30.4	58.3	24.3	28.6	30.9	28.9	50.0	24.9	26.7	13.7
	DRUM	61.3	59.5	33.0	58.6	24.7	28.4	30.8	30.9	50.0	27.1	26.2	16.3
	GraIL	55.4	54.2	27.8	44.3	20.5	20.2	16.5	14.3	42.5	19.9	22.4	15.3
	NBFNet	59.2	57.5	30.4	57.4	19.0	22.9	20.6	18.5	50.0	27.1	26.2	23.3
	RED-GNN	65.3	63.3	36.8	60.6	30.2	<b>38.1</b>	35.1	34.0	52.5	31.9	<u>34.5</u>	<u>25.9</u>
	AdaProp	66.8	64.2	39.6	61.1	19.1	37.2	37.7	35.3	52.2	34.4	33.7	24.7
	DDLRL	<b>70.9</b>	<b>70.2</b>	<b>42.7</b>	<b>67.2</b>	<b>38.2</b>	37.9	<b>39.4</b>	<b>36.1</b>	<b>73.6</b>	<b>35.1</b>	<b>35.0</b>	<b>28.9</b>
	RuleN	73.0	69.4	40.7	68.1	44.6	59.9	<b>60.0</b>	60.5	76.0	51.4	53.1	48.4
	Neural LP	77.2	74.9	47.6	70.6	46.8	58.6	57.1	59.3	87.1	56.4	57.6	53.9
	DRUM	77.7	74.7	47.7	70.2	47.4	59.5	57.1	59.3	87.3	54.0	57.7	53.1
Hit@10 (%)	GraIL	76.0	77.6	40.9	68.7	42.9	42.4	42.4	38.9	56.5	49.6	51.8	50.6
	NBFNet	59.2	57.5	30.4	57.4	19.0	22.9	20.6	18.5	50.0	27.1	26.2	23.3
	RED-GNN	79.9	78.0	52.4	72.1	48.3	62.9	60.3	62.1	86.6	60.1	59.4	55.6
	AdaProp	86.6	83.6	62.6	75.5	55.1	65.9	63.7	63.8	88.6	65.2	61.8	<b>60.7</b>
	DDLRL	<b>98.0</b>	<b>98.8</b>	<b>69.3</b>	<b>95.6</b>	<b>64.5</b>	<b>71.0</b>	<b>63.8</b>	<b>64.5</b>	<b>91.5</b>	<b>69.0</b>	<b>67.6</b>	<u>60.3</u>
	RuleN	63.5	61.1	34.7	59.2	30.9	34.7	34.5	33.8	54.5	30.4	30.3	24.8

DDLRL consistently outperforms its counterparts across various datasets and partition variations, frequently securing the best or second-best rankings. This highlights DDLRL’s capacity to enhance inductive reasoning through a dual denoising approach, effectively filtering out irrelevant paths while retaining those most relevant to the task. For more experimental analysis, please refer to the supplement.

## 5.2 ABLATION STUDIES

**Effectiveness of each component.** To analyze the contributions of each component within DDLRL to its overall performance, we conducted a series of ablation experiments aimed at validating the effectiveness of these model components. Firstly, we investigated the impact of removing the edge-based sampling module, referred to as DDLRL-w.o.-edge. Specifically, we retained only the path-based sampling module. Since the model does not consider the correlations between relations in the knowledge graph (i.e., single rules), inductive reasoning can be affected by the noise introduced by irrelevant edges, potentially resulting in a performance decrease. Secondly, we retained only the path-based sampling module to observe the influence of path-based sampling, denoted as DDLRL-w.o.-path. Due to the model’s limited ability to accurately evaluate reasoning paths, it may be influenced by irrelevant reasoning paths, thereby reducing its generalization capacity. Lastly, to validate the higher credibility of triplet-level single rules, we used entity-level single rules as a reference, referred to as DDLRL-w.o.-triplet. The experimental results are depicted in Table 2. These ablation experiments provide valuable insights into the pivotal roles played by individual components of DDLRL in enhancing its inductive reasoning capabilities.

**Path-Scoring Function.** The path scoring is composed of both the current path score and the remaining path score. To examine the influence of each component, we conducted a simple ablation experiment: "DDLRL-w.o.-current" denotes the removal of the current path score, while "DDLRL-w.o.-remain" denotes the removal of the remaining path score. The experimental results, as shown in Table 3, indicate that the absence of any of these subcomponents leads to a decrease in model performance. This indirectly confirms the importance of both the current path score and the remaining path score.



Table 2: Ablation studies of each component. 'DDLR-w.o.-edge' denotes the model without edge sampling. 'DDLR-w.o.-path' denotes the model without path sampling. 'DDLR-w.o.-triplet' denotes entity-level single rules.

methods	WN18RR(V4)	FB15k237(V4)	NELL-995(V4)
	MRR(%)	MRR(%)	MRR(%)
DDLR-w.o.-edge	73.3	45.2	35.6
DDLR-w.o.-path	76.2	44.6	38.8
DDLR-w.o.-triplet	74.1	43.4	30.8
DDLRL	76.4	46.3	40.6

Table 3: Ablation study results of path-scoring function components. 'DDLR-w.o.-current' refers to the path-scoring functions excluding the current path score. 'DDLR-w.o.-remain' refers to the path-scoring functions excluding the remaining path score.

methods	WN18RR(V2)	FB15k237(V3)	NELL-995(V1)
	MRR(%)	MRR(%)	MRR(%)
DDLRL-w.o.-current	75.9	48.4	77.0
DDLRL-w.o.-remain	75.9	48.2	76.0
DDLRL	79.4	48.8	81.1

**Relevance Between Relations.** To verify the reliability of single rules, we compared them with other methods measuring relation relevance, as shown in Table 4. The results demonstrate a clear advantage of single rules. It also indicates that single rules at the triplet level possess higher credibility, which aids the DDLRL model in filtering out irrelevant edges.

Table 4: Comparisons of different methods to measure the relation relevance. The term "Cosine" represents the computation of cosine similarity between relations, while "KL\_DIV" and "JS\_DIV" respectively denote the computation of KL divergence and JS divergence between relations.

datasets	Ours	Cosine	KL_DIV	JS_DIV
	MRR(%)	MRR(%)	MRR(%)	MRR(%)
NELL-995(V1)	81.1	70.6	64.2	70.3
WN18RR(V1)	79.4	79.3	78.1	78.2

## 6 CONCLUSION

The DDLRL framework proposed in our work effectively enhances knowledge graph completion by integrating dual denoising techniques operating at both the path and edge levels which allows for the efficient filtering and selection of relevant information, resulting in improved accuracy and efficiency in knowledge graph completion tasks. The path-scoring mechanism prioritizes important paths during the reasoning process, reducing the influence of noise from less relevant paths. Furthermore, the edge-based denoising technique further refines the selection of paths. These two components work in concert to enhance the overall performance of the model. Despite the excellent performance, DDLRL has some limitations. Its current evaluation scope is primarily centered on reasoning for missing triplets, a relatively straightforward task. Yet, addressing the complexities associated with multi-hop queries presents a more challenging endeavor. In the forthcoming research, we plan to investigate alternative learning strategies and extend the application of DDLRL to these intricate reasoning

## REFERENCES

Muhammed Fatih Balın and Ümit V. Çatalyürek. (la)yer-neigh(bor) sampling: Defusing neighborhood explosion in gns. 2022.

- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.
- Fan-Yun Chen, Wen-Yu Li, and Shou-De Chang. Fastgcn: fast learning with graph convolutional networks via importance sampling. In *KDD*, pp. 481–490, 2018a.
- Jiajun Chen, Huarui He, Feng Wu, and Jie Wang. Topology-aware correlations between relations for inductive link prediction in knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6271–6278, 2021.
- Jianfei Chen, Jun Zhu, Le Song, and Jian Wang. Adaptive sampling towards fast graph representation learning. In *CIKM*, pp. 1045–1054, 2018b.
- Wei-Lin Chiang, Xuanqing Huang, Pusheng Li, Weikang Yang, Yulei Sui, Siyuan Sun, Runtong Lyu, and Junzhou Huang. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *KDD*, pp. 257–266, 2019.
- Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Conference on Empirical Methods in Natural Language Processing*, pp. 327–336. Association for Computational Linguistics, 2018.
- Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal*, 24(6):707–730, 2015.
- Alberto García-Durán, Mathias Niepert, and Heiner Stuckenschmidt. Kern: kernel embeddings of categorical data. In *International Semantic Web Conference*, pp. 210–218. Springer, 2017.
- William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, pp. 1024–1034, 2017.
- William L Hamilton, Zitao Ying, and Jure Leskovec. Graphsaint: Graph sampling based inductive learning method. In *ICLR*, 2020.
- Vinh Thinh Ho, Daria Stepanova, Mohamed H Gad-Elrab, Evgeny Kharlamov, and Gerhard Weikum. Rule learning from knowledge graphs guided by embedding models. In *The Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part I 17*, pp. 72–90. Springer, 2018.
- Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, pp. 2863–2872. PMLR, 2018.
- Tengfei Ma, Jianfei Chen, Jun Zhu, and Le Song. Decoupling the depth and scope of graph neural networks. In *ICLR*, 2021.
- Sijie Mai, Shuangjia Zheng, Yuedong Yang, and Haifeng Hu. Communicative message passing for inductive relation reasoning. In *AAAI*, pp. 4294–4302, 2021.
- Christian Meilicke, Manuel Fink, Yanjie Wang, Daniel Ruffinelli, Rainer Gemulla, and Heiner Stuckenschmidt. Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion. In *International semantic web conference*, pp. 3–20. Springer, 2018.
- Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems*, 32, 2019.

- Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607. Springer, 2018.
- Chuan Shi, Xiao-Peng Zhang, Zhi-Qiang Cheng, and Yi-Qi Lu. Layer-dependent importance sampling for training deep and large graph convolutional networks. In *AAAI*, pp. 3559–3566, 2019.
- Zhihao Shi, Xize Liang, and Jie Wang. Lmc: Fast training of gnns via subgraph sampling with provable convergence, Feb 2023.
- Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pp. 926–934, 2013.
- Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, pp. 9448–9457. PMLR, 2020.
- Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 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, 2015a.
- Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, and Sujith Choudhury. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 783–793, 2015b.
- Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. 2020.
- Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 417–426, 2018.
- Kai Wang, Siqiang Luo, and Dan Lin. River of no return: Graph percolation embeddings for efficient knowledge graph reasoning. *arXiv preprint arXiv:2305.09974*, 2023.
- Xiang Wang, Xiang Liu, Chao Zhang, Huan Shi, and Xiang Bai. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 950–958. ACM, 2019.
- Wenhan Xiong, Thien Hoang, and William Yang Wang. Deeppath: A reinforcement learning method for knowledge graph reasoning. *arXiv preprint arXiv:1707.06690*, 2017.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pp. 225–233, 2015.
- Carl Yang, Yuxuan Hu, Ruslan Salakhutdinov, and William W Cohen. Performance-adaptive sampling strategy towards fast and accurate graph neural networks. In *ICLR*, 2020.
- Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017.
- Yongqi Zhang and Quanming Yao. Knowledge graph reasoning with relational digraph. In *Proceedings of the ACM Web Conference 2022*, pp. 912–924, 2022.
- Yongqi Zhang, Zhanke Zhou, Quanming Yao, Xiaowen Chu, and Bo Han. Adaprop: Learning adaptive propagation for graph neural network based knowledge graph reasoning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3446–3457, 2023a.

Yuxuan Zhang, Yuanxiang Li, Yini Zhang, Yilin Wang, Yongshen Yang, Xian Wei, and Jianhua Luo. Missing-edge aware knowledge graph inductive inference through dual graph learning and traversing. *Expert Systems with Applications*, 213:118969, 2023b.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34:29476–29490, 2021.