RAW-Explainer: Post-hoc Explanations of Graph Neural Networks on Knowledge Graphs

Ryoji Kubo
ryojikubo@nyu.edu
New York University Abu Dhabi
Abu Dhabi, United Arab Emirates

Djellel Difallah
djellel@nyu.edu
New York University Abu Dhabi
Abu Dhabi, United Arab Emirates

ABSTRACT

Graph neural networks have demonstrated state-of-the-art performance on knowledge graph tasks such as link prediction. However, interpreting GNN predictions remains a challenging open problem. While many GNN explainability methods have been proposed for node or graph-level tasks, approaches for generating explanations for link predictions in heterogeneous settings are limited. In this paper, we propose RAW-Explainer, a novel framework designed to generate connected, concise, and thus interpretable subgraph explanations for link prediction. Our method leverages the heterogeneous information in knowledge graphs to identify connected subgraphs that serve as patterns of factual explanation via a random walk objective. Unlike existing methods tailored to knowledge graphs, our approach employs a neural network to parameterize the explanation generation process, which significantly speeds up the production of collective explanations. Furthermore, RAW-Explainer is designed to overcome the distribution shift issue when evaluating the quality of an explanatory subgraph which is orders of magnitude smaller than the full graph, by proposing a robust evaluator that generalizes to the subgraph distribution. Extensive quantitative results on real-world knowledge graph datasets demonstrate that our approach strikes a balance between explanation quality and computational efficiency.

1 INTRODUCTION

Knowledge graphs (KGs) are heterogeneous networks that capture real-world entities and their relationships. They play a central role in organizing, representing, and integrating structured knowledge across a variety of domains. However, KGs are often incomplete, which makes the task of knowledge completion, i.e., predicting missing links, essential in the field. Recently, Graph Neural Networks (GNNs) have emerged as a powerful approach for KG completion, outperforming traditional rule-based and embedding-based methods [19–21]. However, interpreting the prediction made by GNN-based methods is difficult due to the complex non-linear transformations. Interpretability is critical in knowledge completion applications as it can serve as additional information used by human editors to add facts and to discover the underlying rules and dynamics of KGs.

To address the challenge of GNN explainability, various methods have been proposed for homogeneous graph applications such as node and graph classification [2, 16]. The explanation can be intuitively defined as the most informative subgraph that underlies the model prediction. A popular approach in GNN explainability is the perturbation framework. In this method, informative subgraphs are identified by perturbing the original input graph to a trained GNN in a post-hoc manner [16]. Despite the development of GNN explainability methods for node and graph level tasks on homogeneous graphs, applications to edge-level tasks on heterogeneous graphs such as the KG completion task are limited. A recent study [18] bridges the application of existing methods to knowledge graph tasks by evaluating their performance on synthetic heterogeneous graph datasets with ground truth patterns. However, these explainers' effectiveness on real-world KGs remains unclear as the data often lacks ground truth patterns.

Existing GNN explainability methods face several limitations when applied to heterogeneous link prediction tasks. First, many of these methods were originally developed for node- or graph-level tasks, requiring significant modifications to adapt them for link prediction explanations. Furthermore, most existing approaches are designed for homogeneous graphs and lack support for incorporating heterogeneous information such as relation type. Finally, in the context of KG explanations, providing a connected subgraph is often preferred, as it aligns with the natural patterns present in such datasets and makes it human-interpretable. However, many current methods do not guarantee the connectivity of their generated explanations [18].

Amara et al.[2] argue that an ideal explanation should satisfy both factual and counterfactual requirements. A factual explanation provides sufficient information for the model to predict the true label based on the subgraph, whereas a counterfactual explanation supplies the necessary information for the model to predict the opposite label when the subgraph is removed. Given that knowledge graphs are typically very large and that explanatory patterns are often confined to small paths or connected subgraphs, we focus on uncovering factual explanations that are both interpretable and faithfully capture the behavior of the backbone GNN model. Moreover, an effective explainer should be robust to potential out-of-distribution (OOD) issues when evaluating the informativeness of the subgraph, particularly when these are significantly smaller than the original graph. This challenge arises because the explanatory subgraph may fall outside the distribution of graphs encountered during training, making it difficult to assess its informativeness during the perturbation process [9].

In this work, we propose RAndom Walk guided GNN Explainer for KGs (RAW-Explainer), a novel method designed to address the aforementioned limitations of GNN explainability methods on KG tasks. RAW-Explainer is capable of incorporating heterogeneous information to make explanations for KG link predictions. By utilizing an efficient random walk module, it ensures the connectivity of the output explanations in a scalable manner. Additionally, RAW-Explainer employs a novel distance-based robust evaluator that overcomes OOD issues to assess the informativeness of subgraphs during the perturbation process. We conduct extensive experiments

Table 1: GNN explanation methods and their properties. ✓: Supported. (✓): Requires non-trivial extension. ×: Not supported.

Property	GNN-Exp	PG-Exp	PaGE-Link	PowerLink	**RAW-Exp**
Heterogeneity	×	×	✓	✓	✓
Connectivity	×	×	✓	✓	✓
OOD Robustness	×	×	×	×	✓
Inductive	×	✓	×	×	✓
Scalability	×	✓	×	✓	✓
Link Prediction	(✓)	(✓)	✓	✓	✓

to explore the OOD challenges in existing perturbation-based evaluators, showcasing the benefits of our approach. We also propose a realistic end-to-end evaluation framework for explanations when the ground truth is missing, and apply it to compare state-of-the-art explainers on two real-world KG datasets, empirically validating our claims. The results show that RAW-Explainer yields better explanatory subgraphs under limited edge budgets and is significantly more efficient than existing baselines tailored for the knowledge graph completion task.

In summary, our contributions are:

(1) We propose RAW-Explainer, a novel approach that bridges subgraph-based and path-based GNN explanation methods for knowledge graph completion. By integrating parameterized edge-mask learning with random walks, RAW-Explainer produces connected subgraphs as local explanations that are more interpretable while remaining computationally efficient.
(2) We introduce a robust evaluation framework specifically designed for perturbation-based explainers in knowledge graph completion tasks. Our model evaluator approximates the true graph distribution to improve explanation quality.
(3) We propose a novel evaluation protocol that assesses explanations based on their downstream utility for knowledge graph completion, providing a principled alternative when ground truth explanations are unavailable.

2 RELATED WORK

Graph Neural Networks (GNNs) have emerged as a powerful class of machine learning algorithms for relational graphs. They process a graph along with its node and edge features, iteratively transforming node embeddings through message passing and neighborhood aggregation. Early relational GNN models, such as RGCN [12] and CompGCN [14], perform query-independent transformations. More recent models, including NBFNet [21], RED-GNN [19], and A*Net [20], leverage query-dependent transformations and have demonstrated superior performance in the knowledge graph completion task.

Several GNN explainability methods have been developed to tackle the challenge of interpreting GNN predictions [1, 8, 16]. Table 1 summarizes the properties of several approaches relevant to our method. GNN-Explainer [15] is a pioneering method that explains each node/graph instance by optimizing a mask over the adjacency matrix using mutual information. PG-Explainer [10] learns a parameterized model to output explanations across multiple instances using a factual objective. By parameterizing the explainer model, PG-Explainer can output explanations inductively without needing to be trained on unseen instances at test time. XG-Explainer [9] focuses on alleviating the OOD issue of subgraphs during the perturbation process. It trains an alternative GNN evaluator on an augmented dataset consisting of randomly sampled subgraphs and uses this evaluator in the perturbation process of a base explainer such as PG-Explainer.

In the field of knowledge graph completion explanation, only a handful of methods exist. PaGE-Link [18] is tailored for obtaining explanations for heterogeneous link prediction. It learns a mask over the adjacency matrix for each instance and finds informative paths as explanations for link prediction, optimizing the mask with a factual objective and path-enforcing mask learning. Power-Link [5] employs a graph-powering technique to directly extract informative paths as explanations. Other approaches that explain knowledge graph embeddings, such as KGEx [3], have intrinsic limitations due to their reliance on static embeddings that do not capture the global interactions between entities and relations.

RAW-Explainer extends the benefits of robust evaluation and parameterized global explanation to the case of heterogeneous graphs while drawing inspiration from existing methods to generate local, interpretable subgraphs in a scalable manner.

3 PRELIMINARY

A knowledge graph (KG) is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{E}$ is the set of edges (or triples). Each edge corresponds to a triple (h, r, t), with $h, t \in \mathcal{V}$ and $r \in \mathcal{R}$. The task of KG completion is to identify the missing tail or head entity for a query, i.e., either $(h, r, ?)$ or $(?, r, t)$, such that the triple (h, r, t) holds true in $\mathcal{G}$. By incorporating inverse relations, all queries can be uniformly represented in the form $(h, r, ?)$; for simplicity, we denote a query simply as (h, r).

A GNN model Φ learns the conditional distribution $P_\Phi(\mathbf{Y}|\mathcal{G}, (h, r))$, where $\mathbf{Y}$ is a binary random variable over all of the entities $v \in \mathcal{V}$. For each layer of GNN, the model performs three essential operations: the message operation ψ, the aggregation operation $\bigoplus$, and the update operation ζ. Combining these operations, the l-th GNN layer can be represented as transforming $\mathbf{h}_h^{l-1}$, the node embedding of entity h from the previous layer using Eq. 1.

$$\mathbf{h}_h^l = \zeta\left(\mathbf{h}_h^{l-1}, \bigoplus_{t \in N_h} \psi(\mathbf{h}_h^{l-1}, \mathbf{h}_t^{l-1}, r)\right) \tag{1}$$

where N_h represents the neighborhood of the head entity h, and r is the relation that connects the entities h and t. The initial node embeddings in the first layer are derived from node features, which can

be pre-trained entity embeddings or indicator functions designed to facilitate query-dependent transformations. The final embedding $\mathbf{h}_h^L$, obtained after L layers of computation, is used to compute the score for the entity t as the answer to the query (h, r) through a decoder.

The objective of obtaining GNN explanations varies depending on whether the goal is to obtain factual or counterfactual explanations. When obtaining a *factual explanation* (also referred to as sufficient explanation) for a query (h, r), the task is to find the most informative small subgraph $\mathcal{G}_S \subseteq \mathcal{G}$ such that when the GNN model is fed $\mathcal{G}_S$, it can retain the true label. A generic formulation is given by:

$$\arg\max_{\mathcal{G}_S} \; P_\Phi(\mathbf{Y}|\mathcal{G}_S, (h, r)) \; s.t. \; |\mathcal{G}_S| \leq k \tag{2}$$

where $\mathcal{G}_S$ is the explanatory subgraph, $|\cdot|$ can be defined as the number of edges or nodes, and k is the user-defined budget for the explanatory subgraph. Note that we position ourselves in a *phenomenon*-focus explanation context instead of a *model*-focus explanation context by setting our objective to retain the true label $\mathbf{Y}$ instead of the original prediction $\hat{\mathbf{Y}}$ made by the GNN given $\mathcal{G}$ [2]. This approach enables us to uncover the underlying dynamics and rules of the knowledge graph with an expressive GNN model, rather than merely explaining model behavior. Nevertheless, our framework can be easily converted for the *model*-focus explanation by swapping the objective from $\mathbf{Y}$ to $\hat{\mathbf{Y}}$.

Many perturbation-based explanation methods [10, 15, 17] aim to find the optimal $\mathcal{G}_S$ by maximizing the mutual information between the masked subgraph $\mathcal{G}_S$ and $\mathbf{Y}$. This corresponds to finding the optimal $\mathcal{G}_S$ that satisfies Eq. 2.

4 PROPOSED METHOD: RAW-EXPLAINER

Most GNN explainability methods have been primarily designed for node and graph-level explanations on homogeneous graphs, while the few works that focus on explaining link prediction either aim to identify a restricted path, suffer from scalability issues, or are adapted from homogeneous graph solutions that are often applicable to small graphs. However, knowledge graphs are notoriously large, incomplete, and noisy. We propose RAW-Explainer to address these limitations and to efficiently produce small, explanatory subgraphs. Our approach is based on the following design principles: (1) Ensure the connectivity of the output explanations for better interpretability. (2) Enable inductive explanation capabilities at test time by employing a parameterized model. (3) Accurately evaluate the informativeness of subgraphs during the perturbation process using a robust GNN evaluator. RAW-Explainer achieves these goals through three key components: a heterogeneous mask learner, a random-walk-based connectivity constraint, and a robust GNN evaluator. These components are detailed in the following sections.

4.1 Heterogeneous Mask Learner

We introduce a heterogeneous mask learner that determines the importance of both entities and relations in a knowledge graph. This module learns soft masks for edges to estimate the significance of individual triples. Building on global explanation techniques such as [10], we employ a parameterized model capable of learning edge importance across multiple instances, thereby enabling inductive explanations at test time.

Parametrized Soft-Mask Model. Given a knowledge graph $\mathcal{G}$ and a query (h, r), a pre-trained GNN model Φ is used to compute d-dimensional embeddings for both entities and relations:

$$\mathbf{Z}_E, \mathbf{Z}_R = \Phi(\mathcal{G}, (h, r)), \tag{3}$$

where $\mathbf{Z}_E \in \mathbb{R}^{|\mathcal{V}| \times d}$ contains the entity embeddings and $\mathbf{Z}_R \in \mathbb{R}^{|\mathcal{R}| \times d}$ contains the relation embeddings.

For each edge $(s, p, o) \in \mathcal{E}$, we determine its importance with respect to the query (h, r) by concatenating the embeddings of the source entity s, the relation p, the target entity o, the query head h, and the query relation r. This concatenated vector is then passed through an MLP η:

$$\omega_{s,p,o} = \eta([\mathbf{z}_s; \mathbf{z}_p; \mathbf{z}_o; \mathbf{z}_h; \mathbf{z}_r]). \tag{4}$$

Subsequently, the Gumbel-Softmax function is applied to $\omega_{s,p,o}$ to constrain its value to the interval $[0, 1]$, effectively approximating discrete edge sampling. The collection of all edge scores forms the soft mask Ω of the graph.

Soft-Mask Evaluator. Next, to assess the relevance of the learned mask, we use an evaluation module, a GNN Φ_{eval}, that performs link prediction using the masked graph:

$$\hat{\mathbf{Y}}_\Omega = \Phi_{eval}(\mathcal{G}, (h, r), \Omega). \tag{5}$$

This step assesses how well the mask Ω captures the important edges necessary for answering the query. Although the backbone GNN Φ could be used as the evaluator Φ_{eval}, we explore the limitations of this choice and present an alternative in Section 4.2.

Rather than using the standard GNN update, we integrate a dynamic masking mechanism into each layer. In this framework, the message passed from each neighboring node is scaled by its corresponding mask value. The update rule for node s at layer l is defined as follows:

$$\mathbf{h}_s^l = \zeta\left(\mathbf{h}_s^{l-1}, \bigoplus_{o \in N_s}^{\Omega} \omega_{s,p,o} \cdot \psi(\mathbf{h}_s^{l-1}, \mathbf{h}_o^{l-1}, p)\right). \tag{6}$$

Here, ψ computes the message from neighbor o to node s, and ζ updates the node representation. The operator $\bigoplus^{\Omega}$ denotes a dynamic aggregation that incorporates the mask values. For example, if mean aggregation is used, the normalization term adjusts based on the effective number of neighbors with significant mask values.

4.2 Robust GNN Evaluator

Recent studies have shown that perturbation-based frameworks can be undermined by an out-of-distribution (OOD) issue [7, 9]. In the case of GNNs, this occurs because the explanatory subgraph $\mathcal{G}_S$ can fall outside the original training graph distribution. Hence, the backbone GNN predictor, Φ, trained only on the original graph distribution, may not be able to make accurate predictions to effectively train the learnable mask. Existing solutions create an auxiliary GNN evaluator model, Φ_{eval}, that is trained to approximate the true graph distribution by applying a uniform edge-drop probability on the training graph. This approach simulates potential OOD conditions during training, enabling the GNN evaluator to generalize better when assessing perturbed subgraphs during explanations.

Although the OOD issue has primarily been studied in the context of node and graph classification, the findings strongly suggest that a robust GNN evaluator is just as critical for link prediction. Therefore, we independently explore this claim in Section 5.2, comparing the performance of a standard GNN Φ with a robust GNN evaluator Φ_{eval} for the task of knowledge graph completion.

However, directly applying uniform edge-drop augmentation to knowledge graphs has inherent limitations due to their typically large and sparse structure. Furthermore, the explanatory subgraphs for knowledge graph queries are usually localized and highly relevant to specific entities or relations. Hence, uniform perturbations may introduce irrelevant or overly broad OOD contexts, causing the evaluator to poorly estimate the true graph distribution.

To overcome this, we propose a specialized robust evaluator designed specifically for the evaluation of knowledge graph completion. The proposed evaluator, ϕ_{eval}^{dist}, is trained by assigning edge-dropping probabilities proportional to the edge distance from query entities, thus preserving the contextual relevance and locality of explanatory subgraphs. In our experiments (Section 5.3), we evaluate the effectiveness of ϕ_{eval}^{dist} against a baseline evaluator that uses uniform edge-dropping.

4.3 Structural Constraint

We enforce a structural constraint to ensure that Ω produces a connected subgraph. Since Ω is a soft edge mask that assigns continuous importance scores, to generate the final explanatory subgraph we select the k edges with the highest scores. However, connectivity is crucial for interpretability, as it captures paths from the head to candidate tails via key intermediary nodes. To achieve this, we implement a random-walk assisted approach that rewards edges frequently reached during the walk while penalizing unreachable edges.

To perform the random walk process without compromising the computation efficiency, we perform power iterations of the matrix representation of $\mathcal{G}$ such that the process can be parallelized. Notice that Ω represents a weighted adjacency matrix of $\mathcal{G}$. However $\Omega \in \mathbb{R}^{|\mathcal{V}|^2 \times |\mathcal{R}|}$ due to $\mathcal{G}$ being a multi-directed graph with potentially many edges between entities. We transform Ω to $\Omega' \in \mathbb{R}^{|\mathcal{V}|^2}$ by aggregating the relational dimension. As a design choice, we can choose the aggregation function from *max*, *sum*, *mean*.

Afterward, we transform Ω' to Ω_S using Eq. 7 which is a stochastic adjacency matrix such that the transitional probability from a source entity sums to 1.

$$\Omega_S = (1-\alpha) \cdot \sigma(\Omega') + \alpha \cdot \sigma(\boldsymbol{\tau}) \tag{7}$$

where σ is the softmax function for each source node and $\boldsymbol{\tau} \in \{0,1\}^{|\mathcal{V}|^2}$. $\boldsymbol{\tau}$ represents the teleportation matrix where $\boldsymbol{\tau}_{ij} = 1$ if j is a relevant entity to the query (h, r). We define the relevant entities set ρ from the query head h and the top m predicted tails by Φ_{eval} on the full graph $\mathcal{G}$; these entities serve as both the teleportation targets and initial seeds for the random walk, keeping it focused on the query's key nodes.

Given Ω_S, we can initiate the power iteration process for the random walk. Specifically, we use the Personalized Page Rank (PPR) algorithm [11] such that the obtained node distribution is query-dependent. The initial node distribution $\boldsymbol{\pi}^0 \in \mathbb{R}^{|\mathcal{V}|}$ is defined as $\boldsymbol{\pi}_j^0 = 1/|\rho|$ if $j \in \rho$ and 0 otherwise. We repeat the iteration $\boldsymbol{\pi}^t = \Omega_S \cdot \boldsymbol{\pi}^{t-1}$ until the node distribution converges, i.e. $|\boldsymbol{\pi}^t - \boldsymbol{\pi}^{t-1}| < \epsilon$.

After convergence, we retrieve the top l entities from the final node distribution $\boldsymbol{\pi}^T$. We categorize each edge into two sets:

$$\mathcal{E}_{in} = \left\{(s,p,o) \;\middle|\; (s,p,o) \in \mathcal{E} \;\wedge\; \{s,o\} \subseteq \text{top-}l(\boldsymbol{\pi}^T)\right\},$$

$$\mathcal{E}_{out} = \left\{(s,p,o) \;\middle|\; (s,p,o) \in \mathcal{E} \;\wedge\; \{s,o\} \nsubseteq \text{top-}l(\boldsymbol{\pi}^T)\right\}.$$

Then, we calculate the loss using Equation 8.

$$\mathcal{L}_{PPR}(\Omega) = -\beta_{in} \sum_{e \in \mathcal{E}_{in}} \omega_e + \beta_{out} \sum_{e \in \mathcal{E}_{out}} \omega_e \tag{8}$$

where β_{in}, β_{out} are hyperparameters to control the balance of the weighted average of the edge importance. Using this loss in conjunction with Eq. 5 encourages the explainer to retrieve a connected subgraph explored by PPR as the explanation.

At inference, RAW-Explainer selects the top entities/edges from $\boldsymbol{\pi}^T$ to sample the explanation $\mathcal{G}_S$ that satisfies the user-defined budget, $|\mathcal{G}_S| \leq k$.

4.4 Optimization

We optimize the parameters η such that, for every query (h, r), the learned mask Ω maximizes the masked prediction objective in Eq.5. Intuitively, this process amounts to finding an optimal mask Ω that identifies which edges are essential for reproducing the model's original prediction. Concurrently, we require Ω to satisfy the factual constraint in Eq.2, so that the explanation remains faithful to the predicted fact.

Since we seek to generate concise explanatory subgraphs, we introduce regularization terms that penalize the size of Ω and encourage an entropy-based approximation of discrete sampling. These regularizers push the explainer toward learning a sparse mask, akin to a binary mask rather than a dense or continuous one, following similar techniques in prior work [10, 15, 18]. In practice, the size penalty ensures that only a few edges receive high importance scores, and the entropy penalty nudges mask values to be near 0 or 1, making them easier to interpret.

It is important to note that while we optimize η, the original evaluation model parameters Φ_{eval} remain fixed. If we were to optimize Φ_{eval} jointly with η, the model could adapt its internal representations to rely on edges even when their mask values are low [7]. By freezing Φ_{eval}, we ensure that edge importance truly reflects the features of the current model rather than newly learned dependencies, thus preserving the semantics of the explanation.

4.5 Protocol for Evaluating GNN Explanations

There is currently a lack of consensus on how to evaluate knowledge graph explanations, particularly when the ground-truth is unavailable. A similar challenge arises in node and graph prediction for simpler graphs, where explanation quality is often assessed by measuring the alignment between the explainer's predictions and those of the original GNN. However, this approach is not viable in knowledge graphs due to the large distribution shifts caused by differences between the full KG and the generated explanatory subgraphs.

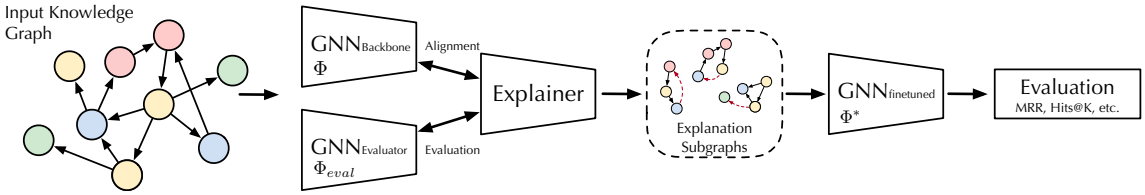


Figure 1: End-to-end evaluation protocol for GNN explanations in knowledge graph completion. Explanations are assessed by fine-tuning the original GNN and evaluating downstream performance on explanation subgraphs.

To address this gap, we propose a downstream performance-based evaluation framework, where the generated subgraphs are used as training datasets to assess their informativeness. Figure 1 illustrates our end-to-end evaluation protocol. Starting with a backbone GNN and a perturbation evaluator, an explainer is trained to generate explanatory subgraphs. The quality of these subgraphs is then assessed by fine-tuning the original GNN using only the extracted subgraphs of a fixed edge size. The fine-tuned model is subsequently evaluated using standard knowledge graph completion metrics such as Mean Reciprocal Rank (MRR) and Hits@K.

The intuition is that a superior explainer captures high-utility subgraphs that retain essential predictive information, yielding higher downstream performance. Conversely, a lower relative performance suggests that the explanations may be missing crucial edges necessary for accurate predictions. This framework offers a systematic and task-specific method for evaluating explanations, establishing a common ground for comparing explainers while ensuring that the outputs are both interpretable and functionally meaningful for knowledge graph completion. Note that while retraining the GNN from scratch is a viable alternative to fine-tuning, it is impractical for instance-based explainers since it requires generating explanations for the entire training dataset.

5 EXPERIMENTS

In this section, we evaluate our proposed RAW-Explainer framework through a series of experiments divided into two sets.

The first set investigates the impact of distribution shift on knowledge graph completion. We begin by examining the performance of the backbone GNN when an increasing number of edges are dropped at random, comparing it to a robust evaluator trained with a uniform edge-dropping probability (Section 5.2). We then evaluate the performance of the GNN on subgraphs surrounding the query head node (i.e., ego networks) at increasing distances and compare these results to those of robust evaluators (Section 5.3). In the second set of experiments, we use our evaluation framework to compare the performance of RAW-Explainer with state-of-the-art baselines (Section 5.4) on two real-world datasets, considering both effectiveness and computational time.

5.1 General Experiment Setup and Datasets

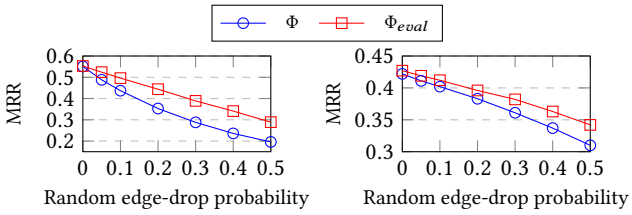
Datasets. In our evaluations, we use two real-world knowledge completion datasets: (1) FB15k-237 [13], a widely-used subset of the Freebase knowledge base containing facts about real-world entities, and (2) WN18RR [6], a lexical knowledge base derived from WordNet, containing semantic relationships among words. FB15k-237 is characterized by higher density, featuring a larger number of edges per node (average node degree of 37.52) compared to WN18RR, which exhibits sparser connections (average node degree of 4.28). To facilitate bidirectional traversal, we augment both datasets with inverse relations, resulting in reverse predicates for each relation.

Table 2: Dataset statistics for knowledge graph completion.

Dataset	#Entity	#Relation	#Train	#Validation	#Test
FB15k-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,134

GNN Backbone. We adopt NBFNet [21], a state-of-the-art GNN architecture for knowledge graph completion. NBFNet operates by learning a differentiable method for pathfinding within the graph, thereby allowing for the modeling of important subgraphs relevant to each prediction, which makes it particularly suitable for studying explainability methods. We follow the hyperparameter configurations detailed in the original paper. Hereafter, the term *backbone GNN* or Φ refers specifically to the NBFNet model trained on our datasets.

Evaluation. For the task of link prediction, where a head query $(h, r, ?)$ or tail query $(?, r, t)$ ranks all nodes in $\mathcal{V}$, we follow the standard filtered ranking protocol [4] and rank the true triple (h, r, t) against all negative triples (h, r, t') or (h', r, t) that do not exist in the KG. We report the mean reciprocal rank (MRR) as a concise evaluation metric, averaged over both head and tail prediction tasks.



(a) WN18RR. **(b) FB15k-237.**

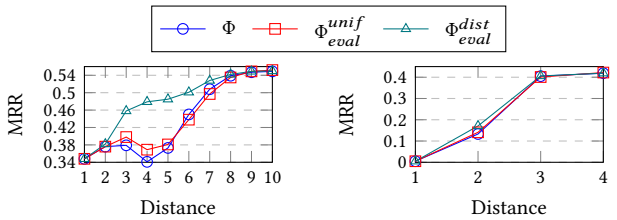
Figure 2: Performance evaluation of Φ, Φ_{eval} on different probability of random edge-drop.

5.2 Distribution Shift GNNs for Knowledge Graphs

In this experiment, we compare two GNN model evaluators: the backbone model Φ, trained exclusively on the full training graph, and an evaluation model Φ_{eval}, trained on the same graph but with a uniform distribution of edges removed. The goal is to examine if the evaluation model that estimates the true graph distribution via random edge dropping provides more robust predictions than the backbone model.

Experimental Setup. Ideally, in scenarios with known ground-truth edge importance, a robust evaluation model should sustain its performance when unimportant edges are removed from the graph $\mathcal{G}$. However, since such ground-truth information is unavailable for real-world datasets, we adopt an alternative experimental strategy. We randomly remove edges from the full graph $\mathcal{G}$ and observe the models' performance. Robustness is assessed based on the evaluator's ability to sustain its performance on modified graphs. We conduct these evaluations on the validation set to preserve the integrity of testing data and to fairly measure model robustness.

Results. Figure 2 illustrates the experimental results. While both models Φ and Φ_{eval} show comparable performance on the full graph $\mathcal{G}$, their performances degrade as the probability of random edge removal increases. This decrease in performance aligns with expectations, as crucial informative edges for accurate predictions may be inadvertently removed during random edge-dropping. However, the evaluation model Φ_{eval} consistently maintains higher accuracy at varying levels of sparsity compared to the backbone model Φ. Given that only a small subset of edges in $\mathcal{G}$ significantly contributes to answering queries [18, 20], random edge removal predominantly affects non-informative edges. The capacity of Φ_{eval} to maintain higher performance under these conditions is critical for accurately evaluating both factual and counter-factual aspects of explanations. This result supports our claim that Φ_{eval} is a more effective and reliable evaluator than Φ in assessing the quality of explanations.



(a) WN18RR. **(b) FB15k-237.**

Figure 3: Performance evaluation of Φ, Φ_{eval}^{unif}, and Φ_{eval}^{dist} on different size of ego networks around the head entity.

5.3 Comparing Variants of Model Evaluators on Distance

In this experiment, we compare three GNN evaluators: the backbone model Φ, an evaluation model $\Phi_{eval}^{\text{unif}}$ trained using a uniform edge-drop distribution, and a distance-biased evaluator $\Phi_{eval}^{\text{dist}}$. Unlike $\Phi_{eval}^{\text{unif}}$, which removes edges uniformly at random, $\Phi_{eval}^{\text{dist}}$ employs a decay bias in edge-dropping, assigning higher removal probabilities to edges that are further away from the head entity. The aim is to determine whether smaller and highly connected subgraphs cause a distribution shift, and which of these models can be utilized as a robust evaluator to train a perturbation-based explainer.

Experimental Setup. A robust evaluator should preserve predictive performance despite changes in subgraph structure, especially in scenarios involving varying distances from the head entity. Similar to the previous experiment, in the absence of ground truth, we adopt the following approach: we vary the radius around the head entity and extract distance-based ego networks for each head entity in our validation set. We then measure each model's MRR at different ego network sizes. Note that the Freebase dataset is dense, and a radius of 4 hops from any node captures a large portion of the graph.

Results. Figure 3 illustrates the results of this experiment on both datasets. Although all three models exhibit comparable accuracy on smaller ego networks, differences emerge as the subgraph diameter increases. The backbone model Φ and the uniform evaluator Φ_{eval}^{unif} demonstrate a performance drop in the case of WN18RR with increasing subgraph radius in the range of 3-5 hops, indicating distribution shifts caused by the removal of distant, informative edges. Unlike the previous experiment, this result is counterintuitive since increasing the radius adds new edges. We ascribe this phenomenon to structural changes that introduce noise for some queries. In contrast, the performance of the distance-biased evaluator Φ_{eval}^{dist} increases monotonically with larger networks and outperforms the other two models. The results for Φ_{eval}^{dist} highlight the utility of using distance-based edge sampling as a training strategy for evaluators, particularly suited for assessing the quality of factually connected GNN-generated explanations.

5.4 Explanations on KG datasets

In this section, we shift our attention to evaluating explanation results following the framework introduced in Section 4.5.

Evaluation. We evaluate the explanations $\mathcal{G}_S$ under different budget constraints. Specifically, we evaluate explanations with budgets $|\mathcal{G}_S| \leq k$, where $|\cdot|$ denotes the number of edges and $k \in \{25, 50, 75, 100, 300, 500\}$. This effectively limits the size of the explanation, enhancing its interpretability.

Baselines. To evaluate the performance of our approach we compare it against various baselines to assess effectiveness.

- GNN-Explainer [15] is a post-hoc, model-agnostic method that provides instance-level explanations for GNN predictions. It operates by optimizing mutual information between predictions and possible explanatory subgraphs.
- PG-Explainer [10] uses a neural network to model edge importance distributions, enabling better generalization and efficiency in inductive settings, and can collectively generate explanations across multiple instances.

- PaGE-Link [18] explores path-based solutions to the GNN explanation problem by generating interpretable explanations for link predictions in heterogeneous graphs through k-core pruning and mask learning techniques.
- Power-Link [5] improves upon PaGE-Link by replacing the shortest-path searching algorithm with a simplified graph-powering technique that can be more efficient and parallelizable.

Experiment Setup. All baselines follow the same protocol, employing the backbone model NBFNet Φ and the evaluator Φ_{eval}^{dist} (see previous section) as components of the perturbation process. Each baseline method generates explanations for validation and test queries. Subsequently, we fine-tune the backbone model using the validation query subgraphs and evaluate the resulting model by using the test set explanation subgraphs. This procedure is designed to distinguish explanation methods that generate useful subgraphs indicated by higher evaluation scores after fine-tuning. The hyperparameters for both the baselines and our method, as well as the best model for each approach, are selected based on the average MRR across all budget constraints on the validation set.

Evaluation Metrics. We report the mean reciprocal rank (MRR) as a concise evaluation metric; alternative metrics indicate similar trends. Moreover, since the KG-specific baselines are instance-based, this poses a significant challenge in generating explanations at scale. Hence we also report the inference time to generate the explanations of the *test* queries.

WN18RR Results. Table 3 presents the results for the WN18RR dataset. RAW-Explainer outperforms all baselines on MRR, demonstrating its ability to extract informative explanations. PG-Explainer achieves the second-best performance; however, PaGE-Link displays competitive performance for smaller budgets. The effectiveness of PaGE-Link in identifying short paths suggests that it excels in simpler relation patterns. Nevertheless, PaGE-Link's performance plateaus at higher budgets due to to the increased complexity for longer paths. Specifically, instance-based methods exhibit substantial disadvantages in inference speed. For instance, despite Power-Link's aim to speedup shortest paths extraction, it remains computationally expensive for collective explanations. Conversely, PG-Explainer is the most efficient method, benefiting from its fast edge selection. However, it requires a high edge budget to form meaningful patterns.

Table 3: Performance evaluation. MRR per subgraph size for various explainers on WN18RR dataset. Best result in bold, second-best result underlined.

	MRR per Subgraph Size (in edges)						Inference Time
	25	50	75	100	300	500	(minutes)
GNN-Explainer	0.445	0.455	0.459	0.462	0.467	0.472	109
PG-Explainer	0.456	0.478	0.494	0.498	0.507	0.520	**6**
PaGE-Link	0.471	0.473	0.472	0.473	0.471	0.472	172
Power-Link	0.473	0.476	0.475	0.476	0.475	0.472	166
RAW-Explainer	**0.482**	**0.500**	**0.504**	**0.508**	**0.518**	**0.521**	17

FB15K-237 Results. Table 4 presents the results on the Freebase dataset. In this experiment, PG-Explainer achieves the highest MRR across all edge budgets. However, due to the graph's density, the resulting subgraphs are highly disconnected (average number of components = 3) hence likely modeling latent information and producing explanatory subgraphs that are not directly interpretable. Closely behind, RAW-Explainer achieves the best performance in local explainability compared to PaGE-Link and Power-Link, while also demonstrating significantly better inference time and interpretability, as it returns a single connected component by design.

Table 4: Performance evaluation. MRR per subgraph size for various explainers on FB15K-237 dataset. Best result in bold, second-best result underlined.

	MRR per Subgraph Size (in edges)						Inference Time
	25	50	75	100	300	500	(minutes)
GNN-Explainer	0.198	0.227	0.243	0.253	0.283	0.294	1663
PG-Explainer	**0.211**	**0.251**	**0.277**	**0.292**	**0.336**	**0.347**	**48**
PaGE-Link	0.203	0.202	0.202	0.202	0.202	0.202	940
Power-Link	0.205	0.203	0.203	0.203	0.202	0.201	9577
RAW-Explainer	0.197	0.232	0.251	0.259	0.299	0.318	266

Results Discussion. The results on both datasets indicate that RAW-Explainer outperforms traditional path-based methods tailored for heterogeneous graphs, while being more efficient at generating collective explanations by focusing on edge masks rather than path-based structures. Meanwhile, methods like PG-Explainer and GNN-Explainer benefit from having access to the entire graph, thereby inferring latent informative features and improving downstream performance. PG-Explainer tends to generate disconnected subgraphs that are difficult to interpret. Overall, RAW-Explainer strikes a balance between efficiency and interpretability by reconciling subgraph interpretability with computational efficiency, and offers a viable tool for knowledge graph completion explanation.

6 CONCLUSIONS

This paper presents RAW-Explainer, a novel framework that generates connected and interpretable subgraph explanations for link prediction in knowledge graphs. By using a neural network to parameterize the generation of a learned edge mask and coupling it with graph powering techniques for generating random walks, RAW-Explainer significantly accelerates the production of collective explanations while yielding highly connected patterns that capture essential predictive information. To overcome the distribution shift between the full KG and the much smaller explanatory subgraphs, we integrate a novel distance based robust evaluator that generalizes effectively to the subgraph distribution. Extensive experiments on real-world datasets demonstrate that RAW-Explainer strikes a balance between generating concise and connected subgraphs that enhance interpretability, effectiveness, and efficiency compared to existing methods.

REFERENCES

[1] Agarwal, C., Queen, O., Lakkaraju, H., Zitnik, M.: Evaluating explainability for graph neural networks. Scientific Data **10**(1), 144 (2023)
[2] Amara, K., Ying, R., Zhang, Z., Han, Z., Shan, Y., Brandes, U., Schemm, S., Zhang, C.: Graphframex: Towards systematic evaluation of explainability methods for graph neural networks. arXiv preprint arXiv:2206.09677 (2022)
[3] Baltatzis, V., Costabello, L.: Kgex: Explaining knowledge graph embeddings via subgraph sampling and knowledge distillation. In: LoG. Proceedings of Machine Learning Research, vol. 231, p. 27. PMLR (2023)

[4] Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS. pp. 2787–2795 (2013)

[5] Chang, H., Ye, J., Lopez-Avila, A., Du, J., Li, J.: Path-based explanation for knowledge graph completion. In: KDD. pp. 231–242. ACM (2024)

[6] Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: AAAI. pp. 1811–1818. AAAI Press (2018)

[7] Jethani, N., Sudarshan, M., Aphinyanaphongs, Y., Ranganath, R.: Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In: AISTATS. Proceedings of Machine Learning Research, vol. 130, pp. 1459–1467. PMLR (2021)

[8] Kakkad, J., Jannu, J., Sharma, K., Aggarwal, C., Medya, S.: A survey on explainability of graph neural networks. IEEE Data Eng. Bull. **46**(2), 35–63 (2023)

[9] Kubo, R., Difallah, D.: Xgexplainer: Robust evaluation-based explanation for graph neural networks. In: Proceedings of the 2024 SIAM International Conference on Data Mining (SDM). pp. 64–72. SIAM (2024)

[10] Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., Zhang, X.: Parameterized explainer for graph neural network. In: NeurIPS (2020)

[11] Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (November 1999)

[12] Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: ESWC. Lecture Notes in Computer Science, vol. 10843, pp. 593–607. Springer (2018)

[13] Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: CVSC. pp. 57–66. ACL (2015)

[14] Vashishth, S., Sanyal, S., Nitin, V., Talukdar, P.P.: Composition-based multi-relational graph convolutional networks. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=BylA_C4tPr

[15] Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. In: NeurIPS. pp. 9240–9251 (2019)

[16] Yuan, H., Yu, H., Gui, S., Ji, S.: Explainability in graph neural networks: A taxonomic survey. IEEE Trans. Pattern Anal. Mach. Intell. **45**(5), 5782–5799 (2023). https://doi.org/10.1109/TPAMI.2022.3204236

[17] Yuan, H., Yu, H., Wang, J., Li, K., Ji, S.: On explainability of graph neural networks via subgraph explorations. In: ICML. Proceedings of Machine Learning Research, vol. 139, pp. 12241–12252. PMLR (2021)

[18] Zhang, S., Zhang, J., Song, X., Adeshina, S., Zheng, D., Faloutsos, C., Sun, Y.: Page-link: Path-based graph neural network explanation for heterogeneous link prediction. In: WWW. pp. 3784–3793. ACM (2023)

[19] Zhang, Y., Yao, Q.: Knowledge graph reasoning with relational digraph. In: WWW. pp. 912–924. ACM (2022)

[20] Zhu, Z., Yuan, X., Galkin, M., Xhonneux, L.A.C., Zhang, M., Gazeau, M., Tang, J.: A*net: A scalable path-based reasoning approach for knowledge graphs. In: NeurIPS (2023)

[21] Zhu, Z., Zhang, Z., Xhonneux, L.A.C., Tang, J.: Neural bellman-ford networks: A general graph neural network framework for link prediction. In: NeurIPS. pp. 29476–29490 (2021)