eXpath: Explaining Knowledge Graph Link Prediction with Ontological Closed Path Rules

Ye Sun
Beihang University
Beijing, China
sunie@buaa.edu.cn

Lei Shi*
Beihang University
Beijing, China
leishi@buaa.edu.cn

Yongxin Tong
Beihang University
Beijing, China
yxtong@buaa.edu.cn

ABSTRACT

Link prediction (LP) is crucial for Knowledge Graphs (KG) completion but commonly suffers from interpretability issues. While several methods have been proposed to explain embedding-based LP models, they are generally limited to local explanations on KG and are deficient in providing human interpretable semantics. Based on real-world observations of the characteristics of KGs from multiple domains, we propose to explain LP models in KG with path-based explanations. An integrated framework, namely eXpath, is introduced which incorporates the concept of relation path with ontological closed path rules to enhance both the efficiency and effectiveness of LP interpretation. Notably, the eXpath explanations can be fused with other single-link explanation approaches to achieve a better overall solution. Extensive experiments across benchmark datasets and LP models demonstrate that introducing eXpath can boost the quality of resulting explanations by about 20% on two key metrics and reduce the required explanation time by 61.4%, in comparison to the best existing method. Case studies further highlight eXpath's ability to provide more semantically meaningful explanations through path-based evidence.

PVLDB Reference Format:
Ye Sun, Lei Shi, and Yongxin Tong. eXpath: Explaining Knowledge Graph Link Prediction with Ontological Closed Path Rules. PVLDB, 18(9): 2818 - 2830, 2025.
doi:10.14778/3746405.3746410

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/cs-anonymous/eXpath.

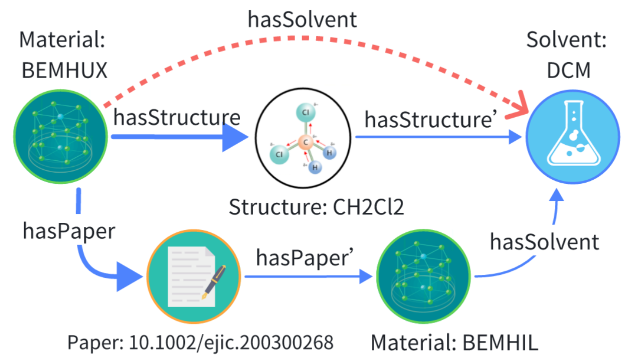

Figure 1: An example of material KG for synthesis route inference. To explain the predicted link ⟨material: BEMHUX, hasSolvent, solvent: DCM⟩ (the dotted red link on the top), two key KG paths (blue links on the middle/bottom) are detected by our method: BEMHUX and DCM share the same sub-structure; BEMHUX, appearing in the same paper with BEMHIL, is aslo synthesized using the DCM solvent. Classical LP explanations (e.g., Kelpie) will select only the single-hop links as explanations (thickened blue links).

1 INTRODUCTION

Knowledge graphs (KGs) [1, 5, 21] commonly suffer from incompleteness, such that link prediction (LP) becomes a crucial task for KG completion, aiming to predict potential missing relationships between entities within a KG. In the deep learning era, advanced KG embedding models (KGE) such as ComplEx [33], TransE [34], and ConvE [10] have been applied to perform the LP task successfully. Yet, due to the inherent black-box nature of deep learning, how to interpret these LP models remains a daunting issue for KG applications. For example, in financial KGs used to make high-stake decisions such as fraud or credit card risk detection, interpretability is required not only for customer engagement purpose [25], but also by the latest law enforcement [9].

Various methods have been developed to interpret the behaviour of LP models, e.g., to explain graph neural network (GNN) based predictive tasks [6, 35, 39], embedding-based models [3, 37], and providing subgraph-based explanations [36, 38, 41]. On KG, the recently proposed adversarial attack methods [3, 28, 31] become a major class of approaches for explaining LP results. The adversarial method captures a minimal modification to KG as an optimal explanation if only a maximal negative impact is detected on the target prediction. In particular, Kelpie [31, 32] introduces entity mimic and post-training techniques to quantify the model's sensitivity to link removal and addition. Despite the success of LP explanation models on KG, they have key limitations in at least two aspects. First, in most methods, only local explanations related to the head or tail entity of the predicted link are considered without exploring the full KG. Second, the explanations generally focus on maximizing computation-level explainability, e.g., the perturbation to predictive power when adding/removing the potential explanation link.

*Corresponding author. Lei Shi is with School of Computer Science and Engineering, Beihang University, and the State Key Laboratory of Complex & Critical Software Environment.

Proceedings of the VLDB Endowment, Vol. 18, No. 9 ISSN 2150-8097.
doi:10.14778/3746405.3746410

They mostly lack semantic-level explainability, which is extremely important for human understanding.

In this work, we are motivated by the real-world observations that corresponds to the limitation of existing LP explanation methods. For instance, in material KGs, experts may prefer path-based explanations—such as shared sub-structures or co-occurrence in the same research paper—over single-hop links, as they capture richer semantics such as causal relationships. Meanwhile, classical methods (e.g., Kelpie) focus on local, entity-centric explanations (e.g., material properties), missing the opportunity to detect multi-hop path explanation. To overcome this limitation, we propose eXpath, a path-based explanation framework that not only suggests minimal KG modifications but also highlights semantically meaningful paths justifying each prediction.

Note that the idea of path-based explanation has also been studied in the recent work of Power-Link [6] and PaGE-Link [39]. However, these works focus on explaining GNN-based embedding models and leveraging graph masks to produce a single explanation capable of assessing numerous KG paths at the same time. In comparison, we consider the explanation by the adversarial attack of factorization-based embedding models, which evaluates only one or a few KG modifications at a time. When generating adversarial explanation, selecting the optimal path from a thousands of candidates poses significant computational challenges. Moreover, another pragmatic challenge lies in the evaluation of path explanation. While the adversarial method works well in quantifying the effectiveness of a single-link explanation, adding/deleting an entire path can lead to substantial KG changes that are difficult to evaluate by the same adversarial method. The contribution of this work is to address the above challenges as summarized below:

- Based on the attributed characteristics of KG, we introduce the concept of relation path, which aggregates paths by their relation types. The explanation analysis then works on the level of relation paths, greatly reducing the computational cost while augmenting the semantics of explanations;
- On the evaluation of path-based explanations, we propose to borrow ontology theory, particularly the closed path rule and property transition rule, which not only reassures the path-based semantics but also guarantees high-occurrence explanations within the whole KG dataset;
- Through extensive experiments across multiple KG datasets and embedding models, we demonstrate the effectiveness of our method, which significantly outperforms existing LP explanation models. Case study also reveals the consistency of path-based explanations with ground-truth semantics.

2 RELATED WORK

2.1 The Explanation of Knowledge Graph Link Prediction (KGLP)

Explainability in Knowledge Graph Link Prediction (KGLP) is a critical research area due to the increasing complexity of models used in link prediction tasks. General-purpose explainability techniques are widely used to understand the input features most responsible for a prediction. LIME [29] creates local, interpretable models by perturbing input features and fitting regression models, while SHAP [19] assigns feature importance scores using Shapley values from game theory. ANCHOR [30] identifies consistent feature sets that ensure reliable predictions across samples. These frameworks have been widely adopted in various domains, including adaptations for graph-based tasks.

GNN-based LP explanation primarily focuses on interpreting the internal workings of graph neural networks for link prediction. Techniques like GNNExplainer [35] and PGExplainer [20] identify influential subgraphs through mutual information, providing insights into node and graph-level predictions, although they are not directly applicable to link prediction tasks. Other methods, such as SubgraphX [36] and GStarX [38], use game theory values to select subgraphs relevant to link prediction. Additionally, PaGE-Link [39] and Power-Link [6] argues that paths are more interpretable than subgraphs and extends the explanation task to the link prediction problem with graph-powering technique. However, these methods aim to identify subsets of the graph (e.g., via weighted masks) that explain predictions in the context of GNN-based models, different from adversarial attack-based explanations.

2.2 Adversarial Attacks on KGE

Adversarial attacks on KGE models have gained attention for assessing and improving their robustness. These attacks focus primarily on providing local, instance-level explanations. The goal is to introduce minimal modifications to a knowledge graph that maximizes the impact on the prediction. Existing approaches fall into two main categories: model-dependent and model-agnostic methods.

Model-dependent methods propose algorithms that approximate the impact of graph modifications on specific predictions and identify crucial changes. Criage [28] applies first-order Taylor approximations for estimating the impact of removing facts on prediction scores. Data Poisoning [3, 37] manipulates embeddings by perturbing entity vectors to degrade the model's scoring function, highlighting pivotal facts during training. ExamplE [16] introduce ExamplE heuristics, which generate disconnected triplets as influential examples in latent space. KE-X [41] leverages information entropy to quantify the importance of explanation candidates and explains KGE-based models by extracting valuable subgraphs. While these methods offer valuable insights, they typically necessitate complete access to the internal mechanics of the model and require extensive theoretical derivations tailored to each architecture.

Recent research has also focused on model-agnostic adversarial attacks, which do not require knowledge of the underlying model architecture and can be applied across architectures. LinkLogic [17] generates path-based explanations by perturbing query triples and using a Lasso regression surrogate model to rank paths based on their contributions. KGEAttack [2] uses rule learning and abductive reasoning to identify critical triples influencing predictions, yet it employs simpler rules and does not consider multiple long rules supporting the facts. Kelpie [31] explains KGE-based predictions by identifying influential training facts, utilizing mimic and post-training techniques to sense the underlying embedding mechanism without relying on model structure. However, these methods are limited to fact-based explanations that focus only on local connections to the head entity (Figure 1's thickened blue links) without capturing the multi-relational context.

2.3 Ontological Rules for Knowledge Graph

Ontological rules for knowledge graphs [11, 12] have been a prominent area of research, as they provide symbolic and interpretable reasoning over knowledge graph data. AMIE [13, 14] and AnyBURL [23, 24] extract rules from large RDF knowledge bases and employ efficient pruning techniques to generate high-quality rules, which are then used to infer missing facts in knowledge graphs. Path-based rule learning has also been explored to improve link prediction explainability. Bhowmik [4] proposes a framework emphasizing reasoning paths to improve link prediction interpretability in evolving knowledge graphs. RLvLR [26, 27] combines embedding techniques with efficient sampling to optimize rule learning for large-scale and streaming KGs. While these methods excel in structural reasoning, they are limited in directly explaining predictions made by embedding-based models.

Recent works have explored the combination of symbolic reasoning with KGE models. For instance, Guo et al.[15] introduced rules as background knowledge to enhance the training of embedding models, while Zhang et al.[40] proposed an alternating training scheme that incorporates symbolic rules. Chudasama et al.[8] enhance explainability by leveraging semantics and causal relationships, improving trust and reliability. Meilicke et al. [22] demonstrated that symbolic and sub-symbolic models share commonalities, suggesting that KGE models may be explained using rule-based approaches. However, these methods have not been directly applied to explain predictions made by KGE models.

3 BACKGROUND AND PROBLEM DEFINITION

3.1 KGLP Explanation

Knowledge Graphs (KGs), denoted as $KG = (\mathcal{E}, \mathcal{R}, \mathcal{G})$, are structured representations of real-world facts, where entities from $\mathcal{E}$ are connected by directed edges in $\mathcal{G}$, each representing semantic relations from $\mathcal{R}$. These edges $\mathcal{G} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, represent facts of the form $f = \langle h, r, t \rangle$, where h is the head entity, r is the relation, and t is the tail entity. Link Prediction (LP) aims to predict missing relations between entities in a KG. The standard approach to LP is embedding-based, where entities and relations are embedded into continuous vector spaces, and a scoring function, $f_r(h, t)$, is used to measure the plausibility of a fact. Evaluation of LP models is typically performed using metrics such as mean reciprocal rank (MRR), which measures how well the model ranks the correct entities when predicting missing heads or tails in the test set $\mathcal{G}_{test}$.

$$MRR = \frac{1}{2|\mathcal{G}_{test}|} \sum_{f \in \mathcal{G}_{test}} \left(\frac{1}{\mathrm{rk}_h(f)} + \frac{1}{\mathrm{rk}_t(f)} \right) \tag{1}$$

where $\mathrm{rk}_t(f)$ and $\mathrm{rk}_h(f)$ represents the rank of the target candidate t in the query $\langle h, r, ? \rangle$ and $\langle ?, r, t \rangle$ respectively.

Understanding the reasoning behind these predictions is essential for model transparency and trust. To address this, explanation methods for embedding-based LP focus on providing instance-level insights into predictions, revealing underlying features like proximity, shared neighbors, or similar latent factors. Since directly perturbing the model's architecture or embeddings is challenging, explanation methods often rely on adversarial perturbations within the training data, such as modifications to the neighborhood of the target triple, to assess the robustness of KGE models.

3.2 Adversarial Attack Problem

Adversarial attacks in the context of KGLP explanations are designed to assess a model's vulnerability to small changes and evaluate the stability of LP models by intentionally degrading their performance through targeted perturbations in the training data. These attacks provide instance-level modifications as adversarial explanations. Given a prediction $\langle h, r, t \rangle$, an explanation is defined as the smallest set of training facts that enabled the model to predict either the tail t in $\langle h, r, ? \rangle$ or the head h in $\langle ?, r, t \rangle$. For example, to explain why the top-ranked tail for $\langle Barack\_Obama, nationality, ? \rangle$ is 'USA', we identify the smallest set of facts whose removal from the training set $\mathcal{G}_{\text{train}}$ would cause the model to change its prediction for $\langle h, r, ? \rangle$ from 'USA' to any entity $e \neq t$, and for $\langle ?, r, t \rangle$ from h to any entity $e' \neq h$. These facts involve the head and tail entities, as they are crucial to the prediction.

We evaluate the impact of the adversarial attack by comparing standard metrics, such as MRR, before and after the attack. Specifically, we train the model on the original training set and select a small subset of the test set $T \subset \mathcal{G}_e$ as target triples for which the model achieves good predictive performance. After removing the attack set from the training set, we retrain the model and measure the degradation in performance on the target set.

Since we focus on small perturbations, the attack is restricted to deleting a small set of triples. To make this process computationally feasible, we adopt a batch mode where the deletion of one target triple may affect others. However, if the triples contain disjoint entities, dependencies between triples are rare and can typically be neglected. The explanatory capability of the attack is measured by the degradation in MRR, defined as: $\delta MRR(T) = 1 - \frac{MRR_{\text{new}}(T)}{MRR_{\text{original}}(T)}$.

3.3 Path-Based Adversarial Explanation

In this work, we focus on path-based adversarial explanations, which integrate rule-based reasoning into adversarial attacks to enhance the interpretability of instance-level modifications. While adversarial attacks identify critical facts by minimally modifying the knowledge graph (KG) to degrade prediction scores, they often lack a clear rationale for why specific facts are deemed critical. We observe that certain KGs, as illustrated in Fig. 1, exhibit semantically meaningful paths that can notably boost the clarity of explanations for individual predictions.

Given a prediction $\langle h, r, t \rangle$, our explanation framework provides the smallest set of training facts that support the prediction, along with a path-based rationale justifying the inclusion of these facts. This rationale is formalized using Closed Path (CP) rules and Property Transition (PT) rules in logical reasoning, which generalize relational patterns from the KG into symbolic, human-interpretable structures. For instance, CP rules (e.g., $r \leftarrow r_1, r_2$) capture multi-hop dependencies, such as inferring a material's solvent usage through shared substructures (Fig. 1). These rules encapsulate causal semantics, grounding adversarial explanations in meaningful relational patterns rather than purely computational perturbations.

Our approach differs significantly from prior path-based explanation methods such as Power-Link [6] and PaGE-Link [39], which

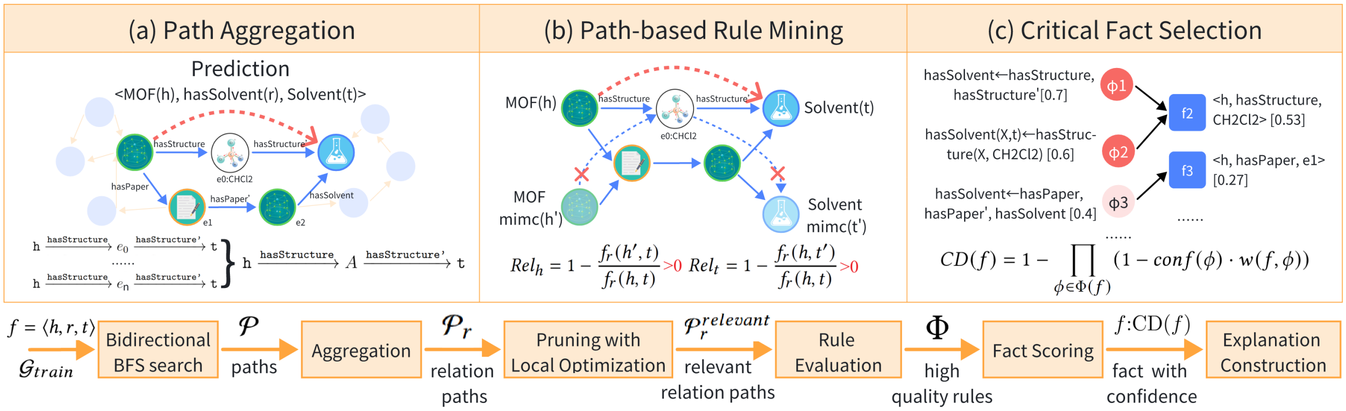

Figure 2: Pipeline of eXpath. (a) *Path Aggregation*: Identifies paths between h and t using bidirectional BFS and aggregate them into relation paths. (b) *Path-based Rule Mining*: Prunes relevant relation paths with local optimization and selects high-confidence closed path (CP) and property transition (PT) rules. (c) *Critical Fact Selection*: Scores candidate facts based on rule weight and confidence, selecting the highest-scoring facts for the final explanation.

focus on learning graph masks as explanations and generating influential paths from these masks. In contrast, adversarial explanation aims to identify minimal modifications to the training set (e.g., removing specific facts) that maximally degrade the model's prediction score, making large-scale dataset modifications infeasible. Moreover, previous path-based explanation method [6, 39] directly utilize original paths, which can yield numerous candidates for each explanation. Exhaustively exploring this vast solution space is computationally challenging. Therefore, eXpath does not directly use paths as explanations. Instead, it enhances the existing adversarial explanation by incorporating path-based rationales to provide semantically meaningful justifications for the modifications.

Our framework caters to two main user categories: domain experts (such as materials scientists, financial analysts) in need of actionable insights consistent with domain-specific reasoning patterns, and data scientists interested in understanding embedding-based models more clearly. Domain experts benefit from path-based explanations (like shared substructures in materials science) that confirm predictions by revealing causal relationships. Data scientists can improve model debugging and trust in predictions by incorporating rules into embedding-based models, thereby connecting symbolic logic with vector-space embeddings.

4 EXPATH METHOD

The eXpath method is designed to explain any given prediction $\langle h, r, t\rangle$ by identifying a small yet effective set of triples whose removal significantly impacts the model's predicted ranking of h and t. Additionally, eXpath provides the rationale for its explanations by presenting the critical paths associated with each selected fact.

The eXpath method follows a three-stage pipeline: path aggregation, path-based rule mining, and critical fact selection. In the path aggregation stage (Figure 2(a)), bidirectional breadth-first search (BFS) is applied to the training facts ($\mathcal{G}_{train}$) to discover paths from h to t, limiting the maximum path length to 3 to ensure interpretability. These paths are then compressed into relation paths ($\mathcal{P}_r$) by removing intermediate entities, reducing the candidate paths while preserving essential semantic structure. In the path-based rule mining stage (Figure 2(b)), we prune the candidate relation paths to retain only the highly relevant ones ($\mathcal{P}_r^{relevant}$) using a local optimization technique based on head and tail relevance. These relevant paths form the body of candidate closed path (CP) rules, evaluated with a matrix-based approach to compute their confidence. Simultaneously, we construct Property Transition (PT) rules from the facts linked to the head and tail entities in $\mathcal{F}^h_{train}$ and $\mathcal{F}^t_{train}$, retaining high-confidence CP and PT rules for fact selection. Finally, in the critical fact selection stage (Figure 2(a)), we score the candidate facts based on the number and confidence of rules they belong to, selecting the highest-scoring facts to form the final explanation.

Notably, while our method efficiently extracts path-based explanations, experiments (Section 5) show that not all KGLP explanations require path-based semantics. In sparser KGs, simple one-hop links can score higher in evaluations. To leverage both approaches, we propose a fusion model that combines eXpath's explanations with those from non-path methods (e.g., Kelpie). By evaluating explanations from both methods, the highest-scoring ones are selected as the final explanation. This fusion model highlights the complementary strengths of different explanation types and demonstrates its potential as a superior overall solution.

4.1 Relation Path and Ontological Rules

When providing path-based explanations for a prediction $f = \langle h, r, t\rangle$, the number of simple paths from h to t grows exponentially with the path length, making even 3-hop paths computationally prohibitive. To mitigate this issue, we focus not on the specific entities traversed by a path but rather on the sequence of relations along the path. This abstraction, referred to as a "relation path," [26] drastically reduces the number of candidate paths while preserving their semantic meaning. By aggregating multiple simple paths into relation paths, we significantly reduce path count while retaining the interpretability crucial for explanations.

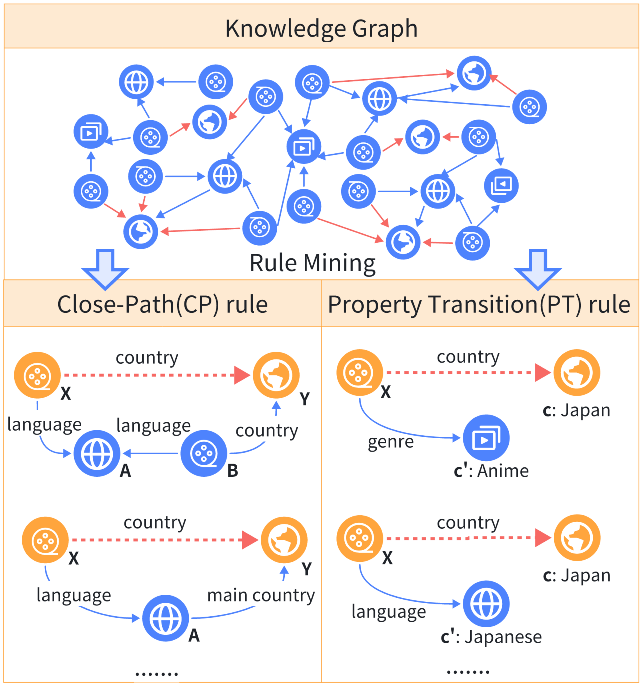

Figure 3: Principles and instances of ontological rules used in our framework. closed path (CP) rules describe the relationship between entities X and Y through alternative paths, while Property Transition (PT) rules capture transitions between different attributes of the same entity. These ontological rules are not predefined but are generalized patterns mined from the knowledge graph, supported by substructures that conform to the specified patterns.

In this study, we introduce two types of rules: Close-Path (CP) rule and Property Transition (PT) rules (illustrated in Figure 3), inspired by the concepts of *binary* and *unary rules with an atom ending in a constant* in AnyBurl [23, 24]. Although PT rules can be converted into CP rules by establishing connections between constants and substituting constants with variables, they remain essential in cases where there is a strong association between two constant entities (e.g., `male` and `female`) that cannot be captured through direct paths. These interpretable rules provide valuable insights into link predictions, laying a robust foundation for generating explanations. Formally, we define two types of rules:

$$
\begin{aligned}
\textbf{CP}: &\quad r\left(A_0, A_n\right) \leftarrow \bigwedge_{i=1}^{n} r_i\left(A_{i-1}, A_i\right) \\
\textbf{PT}: &\quad r\left(X, c\right) \leftarrow r_0\left(X, c'\right) \quad \text{or} \quad r\left(c, Y\right) \leftarrow r_0\left(c', Y\right)
\end{aligned}
\tag{2}
$$

where r and r_i denote relations (binary predicates), A_0, A_i, A_n, X, Y are variables, and c, c' are constants (entities). We use ϕ to denote a rule, where the atoms on the left (h) form the *head* of the rule ($head(\phi)$), and the atoms on the right (r) form the *body* of the rule ($body(\phi)$). To simplify the notation, we use $r \leftarrow r_1, r_2, ..., r_n$ to symbolize CP rules, and relations can be reversed to capture inverse semantics (noted with a single quote, r'). For example, the relation `hypernym(X, Y)` can also be expressed as `hypernym' (Y, X)`.

CP rules are termed "closed paths" because the sequence of relations in the rule body forms a path that directly connects the subject and object arguments of the head relation. This characteristic establishes a strong connection between CP rules and relation paths. Both concepts focus on capturing the structured relationships between entities in a knowledge graph, and their forms are inherently aligned. This alignment allows relation paths to serve as direct candidates for CP rule bodies. In fact, every CP rule can be viewed as a formalized and generalized representation of a relation path, enriched with additional confidence and support.

Moreover, the structured nature of CP and PT rules makes them well-suited for explaining embedding-based predictions, as embedding-based LP models inherently capture the relational graph patterns encoded in CP and PT rules. The alignment between graph patterns and embedding-based models is grounded in their mathematical design. CP rules (e.g., $r \leftarrow r_1, r_2$) in TransE utilize additive operations ($\mathbf{h} + \mathbf{r}_1 + \mathbf{r}_2 \approx \mathbf{t}$) to reflect path composition, while ComplEx employs matrix multiplications ($\mathbf{h} \cdot \mathbf{R}_1 \cdot \mathbf{R}_2 \approx \mathbf{t}$) to capture hierarchical dependencies. Similarly, PT rules (e.g., $\texttt{country}(X, \text{Japan}) \leftarrow \texttt{language}(X, \text{Japanese})$) are based on geometric co-occurrence. These operations ensure that models implicitly learn relational multi-hop chains encoded in CP rules and co-occurrence in PT rules.

To assess the quality of rules, we recall measures used in some major approaches to rule learning [7, 13]. Let ϕ be a CP rule of the form 2. A pair of entities $r\,(e, e')$ satisfies the head of ϕ and there exist entities $e_1, \ldots, e_{n-1}$ in the KG such that $\langle e, r_1, e_1 \rangle, \ldots, \langle e_{n-1}, r_n, e' \rangle$ are facts in the KG, so the body of R are satisfied. Then, the support degree (supp), standard confidence (SC), and head coverage (HC) of ϕ are defined as:

$$
\begin{aligned}
&\text{supp}(\phi) = \#\left(e, e'\right) : \text{body}(\phi)\left(e, e'\right) \wedge r\left(e, e'\right) \\
&SC(\phi) = \frac{\text{supp}(\phi)}{\#\left(e, e'\right) : \text{body}(\phi)\left(e, e'\right)}, HC(r) = \frac{\text{supp}(\phi)}{\#(e, e') : r\left(e, e'\right)}
\end{aligned}
\tag{3}
$$

4.2 Path-based Rule Mining

A critical step for generating path-based explanations is constructing a rule set Φ, which includes both closed path (CP) and Property Transition (PT) rules, as defined in Section 4.1. We do not mine all possible rules across the entire knowledge graph (KG) but instead focus on extracting relevant rules for each prediction from a localized graph relevant to the specific prediction $f = \langle h, r, t \rangle$.

PT rules relevant to a given prediction arise from other facts related to h and t ($f' \in \mathcal{F}^h_{train} \cup \mathcal{F}^t_{train}$). These rules are constructed by replacing common entities in f and f' with variables, which serve as the rule head and body, respectively. For example, for $f = \langle \text{Porco\_Rosso, language, Japanese} \rangle$ and $f' = \langle \text{Porco\_Rosso, genre, Anime} \rangle$, the corresponding PT rule is: $\langle \text{X, language, Japanese} \rangle \leftarrow \langle X, \text{genre, Anime} \rangle$. This rule, similar to the "sufficient scenario" proposed by Kelpie [31], captures whether different entities in the same context satisfy the same prediction.

Calculating metrics for PT rules is relatively straightforward. Based on Equation 3, we simply count the number of facts in $\mathcal{G}_{train}$ that satisfy $\langle X, \text{language}, \text{Japanese} \rangle$ and $\langle X, \text{genre}, \text{Anime} \rangle$ as the

Algorithm 1 Path-based Rule Mining Algorithm

Input: Prediction $f = \langle h, r, t \rangle$, Facts from Training Set $\mathcal{G}_{train}$
Output: Candidate Rule Set for Prediction Φ

$\Phi \leftarrow \emptyset$
{Step 1: CP Rule Extraction}
$\mathcal{P} \leftarrow$ BFSSearch(h, t)
$\mathcal{P}_r \leftarrow$ Aggregation(P)
for each p in $\mathcal{P}_r$ **do**
  $h, h', t, t' \leftarrow$ localOptimization$(f, p, \mathcal{G}_{train})$
  $Rel_h \leftarrow 1 - \frac{f_r(h',t)}{f_r(h,t)}, \quad Rel_t \leftarrow 1 - \frac{f_r(h,t')}{f_r(h,t)}$
  if $Rel_h > 0$ **and** $Rel_t > 0$ **then**
    $(HC, SC, supp) \leftarrow$ RuleEvaluation$(r \leftarrow p, \mathcal{G}_{train})$
    if $SC \geq minSC$ **and** $HC \geq minHC$ **then**
      $\Phi \leftarrow \Phi \cup \{\phi_{CP} : r \leftarrow p[SC \times \frac{supp}{supp+minSupp}]\}$
    end if
  end if
end for
{Step 2: PT Rule Extraction (Take Head PT Rule as Example)}
$\mathcal{F}^h_{train} \leftarrow$ SearchFacts$(h, \mathcal{G}_{train})$
for each $\langle h, r_0, t_0 \rangle$ in $\mathcal{F}^h_{train}$ **do**
  $(HC, SC, supp) \leftarrow$ RuleEvaluation$(r(X,t) \leftarrow r_0(X, t_0), \mathcal{G})$
  if $SC \geq minSC$ **and** $HC \geq minHC$ **then**
    $\Phi \leftarrow \Phi \cup \{\phi_{PT} : r(X,t) \leftarrow r_0(X,t_0)[SC \times \frac{supp}{supp+minSupp}]\}$
  end if
end for
return Φ

head and body counts, respectively. The number of facts satisfying both conditions serves as the support count. Finally, we set a threshold: only rules for which $SC(\phi) >$ minSC and $HC(\phi) >$ minHC are selected to form the PT rule set Φ_{PT}.

CP rules relevant to a prediction, on the other hand, arise from relation paths ($\mathcal{P}_r$) connecting h and t. CP rule mining is more complex than PT rule mining due to the potentially large number of CP rules for a single prediction and the computational expense of evaluating CP rules across the entire knowledge graph. As detailed in Algorithm 1, we first filter $\mathcal{P}_r$ using local optimization, ensuring that only relation paths relevant to the prediction $\mathcal{P}_r^{relevant}$ are considered for evaluation.

During the pruning process, each relation path is assigned a head relevance score and a tail relevance score, which reflect its importance to the prediction. Relation paths with positive head and tail relevance ($Rel_h > 0$ and $Rel_t > 0$) scores are considered relevant to the prediction and retained as candidate rule bodies ($\mathcal{P}_r^{relevant}$) for further evaluation. This filtering approach assumes that a relation path can only serve as a valid rule body if both its head and tail relations are critical to the prediction.

To compute relevance scores, eXpath adopts an local optimization approach inspired by the Kelpie mimic strategy [31]. Mimic entities for the head and tail, denoted as h' and t' (see Fig. 2(b)), are created. These mimic entities retain the same connections as the original head or tail entities, except that all facts associated with the evaluated relation are removed. The embeddings of the mimic entities, along with the original head and tail entities, are then independently trained using their directly connected facts.

Three predictive scores are computed: $f_r(h,t)$, $f_r(h',t)$, and $f_r(h,t')$, where $f_r(h,t)$ represents the model's scoring function for the triple $\langle h, r, t \rangle$. The relevance of a relation is defined as the reduction in the predictive score after removing all facts associated with a specific relation:

$$Rel_h = 1 - \frac{f_r(h',t)}{f_r(h,t)}, \quad Rel_t = 1 - \frac{f_r(h,t')}{f_r(h,t)} \tag{4}$$

Here, Rel_h and Rel_t quantify the importance of relations connected to the head and tail entities. Relative changes in scores are used instead of rank reductions, as scores provide a more robust metric. Rank reductions can be unreliable, especially in local optimization scenarios where mimic entities may overfit, resulting in consistent ranks of 1. This relevance score effectively captures the impact of facts on the prediction by simulating the model's underlying embedding mechanisms.

Finally, eXpath constructs a CP rule set Φ_{CP} for each prediction based on the relevant relation paths $\mathcal{P}_r^{relevant}$ to select high-quality rules that have strong support and confidence. Confidence is computed as $conf(\phi) = SC(\phi) \cdot \frac{supp(\phi)}{supp(\phi)+\text{minSupp}}$, which prevents the overestimation of rules with insufficient support (e.g., $supp < 10$), inadequate for generalizing into a rule. High-confidence CP and PT rules (Φ_{CP} and Φ_{PT}) are retained for fact selection. Strong support and confidence ensure that the selected rules are robust for causal reasoning, enabling eXpath to generate accurate and interpretable path-based explanations.

To efficiently compute metrics for CP rules, we adopt the matrix-based approach from prior work RLvLR [27], which leverages adjacency matrices to verify the satisfiability of rule body atoms. Each relation in the knowledge graph is represented as an $n \times n$ binary adjacency matrix $S(r)$, where entries indicate the presence of corresponding facts. For a CP rule $r \leftarrow r_1, r_2$, the inferred facts are captured by the matrix product $S(r_1) \cdot S(r_2)$, followed by a binarization step to obtain the adjacency matrix $S(r_1, r_2)$. The support, standard confidence (SC), and head coverage (HC) are computed using element-wise logical AND operations and summation over these matrices: support counts overlapping entries between $S(r_1, r_2)$ and $S(r)$, while SC and HC normalize this count by the total inferred or existing r-facts, respectively. This method extends naturally to rules of arbitrary body lengths.

Here we analyze the complexity of Algorithm 1, which consists of the following components: (1) the bidirectional BFS search for generating candidate paths, with a complexity of $O(d^{\frac{L}{2}})$, where $d = \frac{2M}{N}$ is the average node degree, L is the maximum path length, and N, M, and R denote the number of nodes, edges, and relations respectively.; (2) path aggregation, which merges paths by relation sequences and results in $|\mathcal{P}_r| = O(\min\{d^{\frac{L}{2}}, R^L\})$; (3) local optimization, which trains on a subgraph of size $O(d)$ with a complexity of $O(dT)$, where T is the model-specific training cost (e.g., $O(D)$ for TransE and $O(D^2)$ for RESCAL, where D represents the dimension of embeddings); and (4) rule evaluation, which scans training facts with a complexity of $O(ML)$ for L-hop paths. Thus, the overall complexity is $O(\min\{R^L, d^{\frac{L}{2}}\} \cdot (ML + dT))$, dominated by $O(M)$, as L, d, R and T are limited by dataset characteristics or predefined bounds. This linear scalability facilitates the effective application of large-scale KGLP tasks' explanations.

4.3 Critical Fact Selection

This section details the method for selecting an optimal set of facts to explain a given prediction triple $\langle h, r, t\rangle$ leveraging the rules extracted in the previous step. The core idea is to identify the most critical fact or a combination of facts within the paths connecting the head and tail entities. Each fact is scored based on its contribution to the prediction and the final explanation set is constructed by selecting the highest-scoring facts.

Several key factors are taken into account to determine the significance of a fact: (1) Facts that satisfy a larger number of rules are given higher priority, as this indicates their broader relevance within the prediction. (2) Rules with higher confidence are weighted more heavily, reflecting their more robust causal support. (3) The frequency and position of a fact within a rule also play a role; facts appearing more frequently and in critical positions (e.g., adjacent to the head or tail entity) are considered more important.

To model the contribution of a fact that satisfies multiple rules, we adopt a confidence degree (CD) aggregation approach inspired by rule-based link prediction methods [26]. The CD of a fact f is calculated using the confidence values of all the rules that infer f in a Noisy-OR manner. we define the CD of f as follows:

$$CD(f) = 1 - \prod_{\phi \in \Phi(f)} (1 - conf(\phi) \cdot w(f, \phi)) \tag{5}$$

where $\Phi(f)$ is the set of rules inferred from the prediction, $conf(\phi)$ is the confidence of rule ϕ, and $w(f, \phi)$ represents the importance of fact f within rule ϕ, calculated based on the weighted frequency:

$$w(f, \phi) = \frac{Rel_h(\phi) \cdot p_h(f, \phi) + Rel_t(\phi) \cdot p_t(f, \phi)}{Rel_h(\phi) + Rel_t(\phi)} \tag{6}$$

where $Rel_h(\phi)$ and $Rel_t(\phi)$ are the relevance scores of the rule's head and tail relations, respectively. The term $p_{h/m/t}(f, \phi)$ represents the frequency of f 's appearances in the head/middle/tail of all paths related to rule ϕ. This formulation ensures that facts appearing more prominently in rules are scored higher.

According to Equation 6, only facts that are adjacent to the head or tail are considered, while non-adjacent facts are disregarded. This selection is guided by two principles: (1) Embedding sensitivity ensures that adjacent facts (e.g., $\langle h, r_1, A\rangle$) primarily impact the embeddings of h or t, while intermediate facts have weaker effects. (2) An empirical analysis on FB15k-237 illustrates that head/tail-adjacent facts show significantly higher mean contribution ($\overline{p}_h$ = 0.0217, $\overline{p}_t$ = 0.0037) compared to non-adjacent facts ($\overline{p}_m$ = 0.0005). Although middle facts may occasionally contribute (with only 0.2% of facts having $p_m > 0.01$), their influence is overshadowed by head/tail facts ($p_m \ll p_h, p_t$). This suggests that adjacent facts are more likely to be shared among multiple paths within a rule, making them more critical for explaining the prediction.

In PT rules, the importance score for a fact $w(f, \phi)$ is simplified to 1, as the rule corresponds to a unique fact for a given prediction.

After assigning each candidate fact a CD score, we rank all candidate facts by their scores and select the highest-ranked facts as the explanation. This approach ensures that the selected facts are those most strongly supported by high-quality, relevant rules, providing robust and interpretable explanations for the given prediction.

Table 1: Statistics of benchmark datasets.

KG Dataset	Entities	Relation Types	Train Facts	Valid Facts	Test Facts
FB15k	14,951	1,345	483,142	50,000	50,971
FB15k-237	14,541	237	272,115	17,535	20,466
WN18	40,943	18	141,442	5,000	5,000
WN18RR	40,943	11	86,835	3,034	3,134

5 EXPERIMENT

5.1 Experimental Setup

We evaluated eXpath on the knowledge graph link prediction task using four benchmark datasets: FB15k and FB15k-237 [18] (derived from Freebase), and WN18 and WN18RR [4] (based on WordNet). As provided in Table 1, As provided in Table 1, FB15k, built from FreeBase (a real-world knowledge base), includes relations like born-in and part-of, but its test data contained reversed relationships, making prediction tasks artificially easy. This led to FB15k-237, a revised version that removes these reversed links. Similarly, WN18, based on WordNet (a semantic network), models linguistic relations like hypernym (e.g., cat is a feline) but suffered from the same flaw. Its improved version, WN18RR, excludes reciprocal relations to ensure fairer evaluation. We followed standard dataset splits and maintained identical training parameters before and after fact removal to ensure consistency across comparisons.

We compared the performance of eXpath against five contemporary methods dedicated to LP interpretation: Kelpie [31], Data Poisoning (DP) [37], Criage [2], KE-X [41], and KGEAttack [2]. These implementations are publicly available, and we tailored the code sourced from their respective Github repositories. Since the explanation framework is compatible with any Link Prediction (LP) model rooted in embeddings, we conduct experiments on three models with different loss functions: CompEx [33], ConvE [10], and TransE [34]. To ensure fairness between the explanation methods, we restrict the number of facts that can be removed. Specifically, DP, Criage, KGEAttack limit the removal to at most one fact, whereas KE-X, Kelpie and eXpath can remove one or four facts. Based on experiments and existing literature, we set the thresholds minSC = 0.1, minHC = 0.01, minSupp = 10. These parameters are adapted from the definitions of high-quality rules in prior work [13].

To evaluate the effectiveness of adversarial explanations, we rigorously follow the protocol established in prior work (e.g., Kelpie, KGEAttack). The evaluation process begins by constructing an evaluation set $T \subset \mathcal{G}_e$, which consists of 100 triples selected from the test set. These triples are chosen based on the original model's predictive performance, requiring a reciprocal rank ($RR(M_o, f)$) greater than 0.5 to ensure each triple is correctly predicted with at least one head or tail rank being 1. This selection criterion guarantees high-quality predictions while maintaining practical applicability, as overly strict criteria (e.g., requiring both head and tail ranks to be 1) would unnecessarily limit the scope of evaluable scenarios.

The adversarial attack process involves removing critical facts identified by explanation methods from the training set. All 100 triples are attacked simultaneously, and the model is retrained once after removing triples in explanations of all predictions. This batch approach aligns with prior work (e.g., Kelpie and KGEAttack) to avoid computational overhead from repeated retraining. To minimize dependencies, triples are selected to have disjoint head/tail entities, ensuring minimal overlap in entities or relations.

The model's explanatory capability is quantified using two metrics: the relative reduction in Hits@1 ($\delta H@1$) and Mean Reciprocal Rank (δMRR). These metrics compare the performance of the retrained model M_x (after fact removal) against the original model M_o, with δMRR prioritized for its robustness to rank fluctuations. While $\delta H@1$ measures the drop in top-ranked predictions, its sensitivity to training stochasticity—particularly in fragile models like TransE—makes δMRR, which aggregates rank positions across all candidates, a more stable indicator of explanation quality. The metrics are formally defined as:

$$
\begin{aligned}
H@1(M_x, f) &= \frac{1}{2}(\mathbf{1}(rk_h(M_x, f) = 1) + \mathbf{1}(rk_t(M_x, f) = 1)) \\
\delta H@1(M_x, T) &= 1 - \frac{\sum_{f \in T} H@1(M_x, f)}{\sum_{f \in T} H@1(M_o, f)} \\
\delta MRR(M_x, T) &= 1 - \frac{\sum_{f \in T} RR(M_x, f)}{\sum_{f \in T} RR(M_o, f)}
\end{aligned}
\tag{7}
$$

where $\mathbf{1}(\cdot)$ is an indicator function, and $RR(M_x, f)$ computes the average of reciprocal ranks for head and tail predictions. The stochasticity of model training and small dataset size (100 predictions) can cause significant variability in $\delta H@1$ values. This issue is exacerbated for fragile models like TransE, where ranks fluctuate even without attacks. We address this by averaging results over five experimental runs. Each embedding model (e.g., ComplEx, ConvE, TransE) uses a distinct subset of 100 triples customized to its predictive capabilities. This is because a triple correctly predicted by one model may not yield satisfactory results on another model.

We also evaluate fusion methods (e.g., Kelpie + eXpath) by selecting the explanation that yields the greater reduction in metric between Kelpie and eXpath. Taking δMRR as an example, for each fact f to be explained, we define the reciprocal rank of the combined method as $RR(M_{x+y}, f) = \min(RR(M_x, f), RR(M_y, f))$. The overall metric for the fusion method is then calculated using the equation 7. By selecting the minimum value between the two methods, the fusion method enhances explanation performance.

5.2 Explanation Results

Tables 2 and 3 demonstrate the overall effectiveness of the eXpath method in generating LP explanations, evaluated using the $\delta H@1$ and δMRR metrics as defined in Equation 7. For a fair comparison, explanation methods are categorized based on explanation size (i.e., the number of facts provided). The first section of each table (top 9 rows) presents results for 6 single-fact explanations (L1) and their fusion models, such as Criage, KE-X, DP, Kelpie, KGEAttack, and eXpath, which offer one fact per explanation. The second section (bottom 4 rows) shows results for four-fact explanations (L4), including KE-X, eXpath, Kelpie, and their fusion.

For single-fact explanations, eXpath achieves the best average performance, with an average of 0.611 in $\delta H@1$ and 0.494 in δMRR. KGEAttack performs comparably, reaching an average of 0.585 in

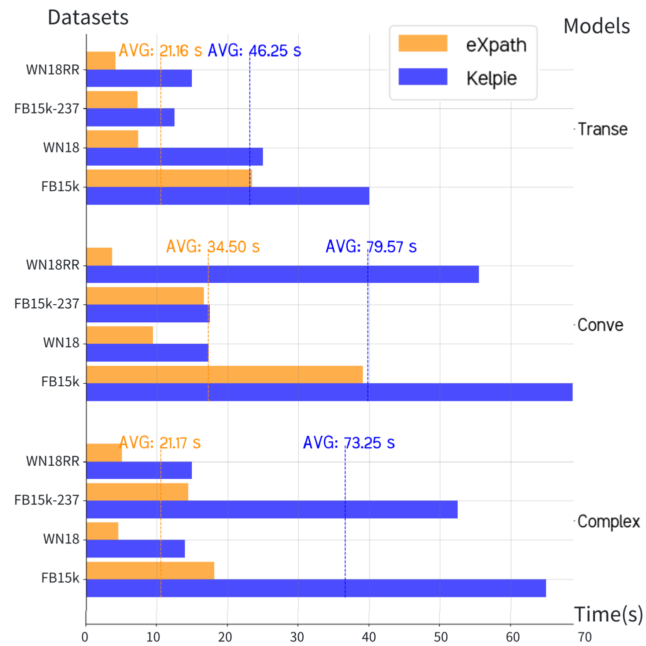

Figure 4: Average times in seconds to extract an explanation for Kelpie and eXpath.

$\delta H@1$ and 0.492 in δMRR. Both methods significantly outperform Criage and Kelpie, surpassing them by at least 15.4% in $\delta H@1$ and 23.6% in δMRR on average. Notably, eXpath secures at least the second-best performance in 20 out of 24 settings and significantly outperforms all methods in 12 settings. Interestingly, eXpath explanations exhibit dataset-specific preferences. Compared to KGEAttack, eXpath performs better in explaining relation-dense datasets such as FB15k-237, achieving an average improvement of 50.3% in $\delta H@1$ and 43.8% in δMRR. On other datasets, the performance of both methods is similar.

In a more practical four-fact scenario, only eXpath and Kelpie support multiple facts as explanations. eXpath, which directly selects the top-scoring set of up to four facts, outperforms Kelpie in 22 out of 24 settings with statistical significance (p-value < 0.05) across five runs. Specifically, eXpath achieves an average of 0.785 in $\delta H@1$ and 0.663 in δMRR, while Kelpie achieves averages of 0.691 in $\delta H@1$ and 0.590 in δMRR. Notably, four-fact explanations of eXpath consistently outperform single-fact explanations across all settings, emphasizing the importance of multi-fact combinations for meaningful explanations. This is particularly evident in dense datasets like FB15k and FB15k-237, where four-fact explanations show an average improvement of 69.5% in $\delta H@1$ and 87.7% in δMRR, compared to single-fact explanations. In contrast, for sparser datasets like WN18 and WN18RR, the improvements are more modest, with average gains of 11.3% in $\delta H@1$ and 41.4% in δMRR. Dense graphs, such as FB15k, contain many synonyms or antonyms for relations (e.g., `actor-film`, `sequel-prequel`, `award-honor`), meaning that even if one fact is removed from an explanation, other related facts remain in the knowledge graph, making adversarial attacks less

Table 2: $\delta H@1$ comparison across different models and datasets using various explanation methods. All results are averaged over five runs, with higher values indicating better performance. The original $H@1$ is 1 for all candidate predictions ($H@1 > 1$ predictions are excluded). Methods with "+eXpath" indicate fusion approaches that combine the given method with eXpath.

Max Exp. Size	Method	ComplEx				ConvE				TransE				AVG
		FB15k	FB15k-237	WN18	WN18RR	FB15k	FB15k-237	WN18	WN18RR	FB15k	FB15k-237	WN18	WN18RR	
single -fact exp.	Criage [28]	.087	.105	.080	.203	.153	.162	.270	.256	—	—	—	—	.165
	KE-X [41]	.035	.153	.141	.379	.102	.152	.234	.318	.174	.292	.493	.384	.238
	DP [37]	.529	.315	.799	.758	.246	.162	.794	.829	.304	.326	.910	.709	.557
	Kelpie [31]	.576	.395	.578	.593	.229	.222	.567	.667	.261	.281	.792	.779	.495
	KGEAttack [2]	.547	.290	.829	.764	.237	.212	.929	.915	.365	.213	.938	.779	.585
	eXpath	.512	.395	.834	.797	.271	.343	.929	.891	.313	.337	.938	.767	.611
	DP+eXpath	.570	.500	**.859**	.813	.331	.414	**.936**	**.946**	.374	**.438**	**.944**	.826	.663 (+19%)
	Kelpie+eXPath	**.657**	**.540**	**.859**	**.835**	**.364**	**.424**	.929	.915	**.417**	.427	**.944**	**.872**	**.682** (+38%)
	KGEA.+eXpath	.576	.452	**.859**	.802	.322	.384	.929	**.946**	**.417**	.360	.938	**.872**	.655 (+12%)
four -fact exp.	KE-X	.145	.177	.603	.841	.102	.141	.511	.589	.235	.281	.632	.430	.391
	Kelpie	.767	.581	.829	.940	.534	.303	.816	.946	.374	.427	.868	.907	.691
	eXpath	.802	.661	.920	.951	.542	.566	.957	.984	.539	.573	**.965**	**.965**	.785
	Kelpie+eXpath	**.831**	**.742**	**.935**	**.989**	**.653**	**.596**	**.965**	**.984**	**.609**	**.674**	**.965**	**.965**	**.826**

Table 3: δMRR comparison across different models and datasets using various explanation methods. All results are averaged over five runs, with higher values indicating better performance. The original MRR is above 0.5 in all candidate predictions.

Max Exp. Size	Method	ComplEx				ConvE				TransE				AVG
		FB15k	FB15k-237	WN18	WN18RR	FB15k	FB15k-237	WN18	WN18RR	FB15k	FB15k-237	WN18	WN18RR	
single -fact exp.	Criage	.045	.051	.058	.163	.024	.031	.157	.150	—	—	—	—	.085
	KE-X	.007	.072	.121	.306	.023	.017	.132	.194	.039	.104	.283	.279	.131
	DP	.451	.187	.729	.668	.140	.058	.728	.785	.157	.141	.742	.613	.450
	Kelpie	.457	.238	.491	.483	.123	.076	.514	.578	.075	.115	.700	.664	.376
	KGEAttack	.463	.172	.766	.684	.159	.104	.889	.853	.190	.091	.877	.659	.492
	eXpath	.430	.233	.774	.688	.183	.130	.889	.810	.159	.165	.877	.596	.494
	DP+eXpath	.491	.282	**.803**	.711	.241	.211	**.900**	**.893**	.239	**.252**	.891	.675	.549 (+22%)
	Kelpie+eXpath	**.534**	**.309**	.795	**.718**	**.245**	.206	.895	.848	.225	.239	**.893**	**.734**	**.553** (+47%)
	KGEA.+eXpath	.495	.262	.799	.712	.239	**.215**	.889	.883	**.261**	.223	.877	.723	.548 (+12%)
four -fact exp.	KE-X	.086	.087	.544	.771	.031	.055	.464	.490	.105	.109	.471	.307	.293
	Kelpie	.632	.434	.777	.891	.391	.143	.795	.919	.203	.199	.805	.893	.590
	eXpath	.680	.452	.875	.887	.366	.327	.924	.952	.354	.261	.937	.943	.663
	Kelpie+eXpath	**.718**	**.519**	**.900**	**.941**	**.468**	**.401**	**.949**	**.966**	**.406**	**.332**	**.952**	**.960**	**.709**

effective. This observation highlights the need for multi-fact explanations to fully capture the predictive context.

The fusion methods (e.g., Kelpie+eXpath), combining eXpath(L1) with DP, Kelpie(L1), and KGEAttack improves δMRR by 22%, 47%, and 12%, respectively. The eXpath-Kelpie fusion improves Kelpie alone by 20%. The results demonstrate that the path-based explanations of eXpath offer unique insights and complementary perspectives that differ significantly from those provided by other adversarial methods, particularly when combined with Kelpie. We also notice that L1 fusion methods converge to an upper bound ($\delta MRR \leq 0.56$, $\delta H@1 \leq 0.69$), indicating that single-fact explanations have inherent limitations. Multi-fact approaches are necessary for satisfactory explanations in link prediction tasks.

In terms of efficiency, Figure 4 compares the average explanation time per prediction between eXpath and Kelpie. eXpath achieves significantly faster explanation speeds, averaging 25.61 seconds per

Table 4: δMRR between different models and datasets with different Fact Position Preferences (Rows 1-6) and Rule Component Ablations (Rows 7-12). Top section compares `all` (unrestricted), `head` (head-related), and `tail` (tail-related) fact position settings. Bottom section evaluates the impact of excluding CP rules (w/o CP) and PT rules (w/o PT).

Max Exp. Size	Method	ComplEx				ConvE				TransE				AVG
		FB15k	FB15k-237	WN18	WN18RR	FB15k	FB15k-237	WN18	WN18RR	FB15k	FB15k-237	WN18	WN18RR	
1	eXpath(all)	.431	.233	.774	.696	.163	.135	.889	.833	.159	.149	.877	.406	.479
	eXpath(head)	.433	.243	.774	.693	.165	.119	.889	.810	.148	.127	.877	.598	.490
	eXpath(tail)	.418	.125	.759	.635	.147	.088	.889	.787	.159	.071	.877	.000	.413
4	eXpath(all)	.680	.453	.807	.878	.370	.319	.900	.939	.355	.270	.918	.826	.643
	eXpath(head)	.659	.438	.877	.887	.372	.290	.925	.952	.346	.271	.935	.942	.658
	eXpath(tail)	.630	.227	.833	.818	.324	.103	.877	.859	.232	.135	.843	.125	.501
1	eXpath	.431	.223	.774	.693	.163	.135	.889	.810	.159	.149	.877	.598	.492
	eXpath (w/o CP)	**.276**	.195	**.757**	**.659**	**.083**	**.125**	**.448**	**.423**	**.106**	**.153**	**.520**	**.574**	**.360** (-27%)
	eXpath (w/o PT)	.431	**.118**	.774	.685	.154	.047	.889	.853	.155	.097	.877	.558	.470 (-4.5%)
4	eXpath	.680	.453	.877	.887	.370	.319	.925	.952	.355	.270	.935	.942	.664
	eXpath (w/o CP)	**.477**	.416	**.875**	.877	**.212**	.295	**.708**	**.835**	**.190**	**.276**	**.800**	**.936**	**.575** (-13.5%)
	eXpath (w/o PT)	.622	**.305**	.833	**.839**	.341	**.159**	.925	.953	.329	.174	.941	.930	.613 (-6%)

prediction, which is approximately 38.6% of Kelpie's average time of 66.36 seconds. This efficiency is attributed to eXpath's localized optimization within relation groups and its straightforward scoring-based fact selection process, compared to Kelpie's exhaustive traversal of connections and time-intensive combinatorial searches.

In conclusion, eXpath demonstrates clear advantages in both performance and execution efficiency, highlighting its potential as a robust framework for path-based adversarial explanation.

5.3 Fact Position Preferences

Many adversarial methods (e.g., KE-X [41], DP [37], and Kelpie [31]) typically select facts directly connected to head entities (head-related facts) for explanations. To further evaluate this preference, we analyze the impact of fact position using three settings: `all` (unrestricted position), `head` (head-related facts), and `tail` (tail-related facts). Results in Table 4 reveal that the `head` setting (L1: 0.490 / L4: 0.658) outperforms the `all` setting (L1: 0.479 / L4: 0.643) on average, and both settings consistently surpass the `tail` setting (L1: 0.413 / L4: 0.501). The `tail` setting consistently weakens performance across all datasets, with significant drops in FB15k-237 (-50%) and WN18RR (-40%) compared to the `head` setting. These results validate the effectiveness of selecting head-related facts, as seen in other adversarial methods. Empirical analysis of node degree distributions reveals that tail entities generally exhibit higher degrees than head entities, making it challenging for traditional adversarial methods to select tail-related facts. While these methods inherently favor head-related facts, such constraints may limit the diversity and semantic richness of explanations.

Dataset characteristics significantly influence the effectiveness of fact position restrictions. For FB15k and FB15k-237, the `all` setting (L1: 0.212 / L4: 0.408) generally outperforms the `head` setting (L1: 0.206 / L4: 0.396), while for WN18 and WN18RR, the `all` setting (L1: 0.746 / L4: 0.878) notably underperforms compared to the `head` setting (L1: 0.774 / L4: 0.920). A possible reason is that FB15k and FB15k-237 are dense graphs, encouraging models to balance head and tail entity modeling. In sparser datasets like WN18RR, head entities often represent concepts with a few relations (average degree < 5), while tail entities serve as hubs with numerous relations (average degree > 100), making head-related facts far more impactful than tail-related facts. Based on these observations, we apply fact position restrictions based on graph density (average degree). In this paper, we apply head-related restrictions for low-density datasets (average degree < 20, e.g., WN18 and WN18RR) and unrestricted selection for high-density datasets (average degree > 20, e.g., FB15k and FB15k-237).

5.4 Ablation Study on Rule Components

To evaluate the individual contributions of CP and PT rules, we conducted ablation experiments by independently removing each rule type during fact scoring. The results (Table 4) reveal distinct roles for these components: CP rules dominate in modeling multi-hop relational patterns, with their removal causing a 13.5%~27% average δMRR drop, while PT rules enhance explanation diversity through property correlation, showing a 4.5%~6% average δMRR drop when excluded. This divergence highlights CP rules as the core mechanism for capturing semantic dependencies, whereas PT rules act as complementary validators of co-occurrence patterns.

Dataset-specific analyses reveal distinct rule dominance patterns. In FB15k, CP rules prove indispensable (38% δMRR drop when removed), excelling at path-based reasoning such as film sequel/prequel relationships (e.g., rule (2) ~ (6) in Figure 5(b)). ConvErsely, PT rules dominate in FB15k-237 (42% δMRR drop when removed), where sparse relations rely on their ability to validate indirect correlations like language-country mappings (`country(X,`

Table 5: Comparison of explanations generated by five adversarial methods for three representative examples. Each cell contains the δMRR in the first row, followed by the explanation sets generated by each model.

Prediction	Criage	Data Poisoning	KGEAttack	Kelpie	eXpath
(1) e_2, award_nominee, e_1 (from complex FB15k)	[0.00] Joan_Allen, award, e_2	[0.89] e_1, award, e_2	[0.89] e_1, award, e_2	[L1: 0.25/L4: 0.38] e_2, award_nominee, Anna_Paquin e_2, award_nominee, Shohreh_Aghdashloo e_2, award_nominee, Julia_Ormond e_2, award_nominee, Amanda_Plummer	[L1: 0.89/L4: 0.95] e_1, award, e_2 e_2, award_nominee, Joan_Allen Tony_Award.., award_nominee, e_1 Academy_Award.., award_nominee, e_1
(2) Porco_Rosso, country, Japan (from conve FB15k)	[0.50] Walt_Disney.., film, Porco_Rosso	[0.48] Porco_Rosso, edited_by, Hayao_Miyazaki	[0.00] Anime, films_in_this_genre, Porco_Rosso	[L1: 0.62/L4: 0.74] Hayao_Miyazaki, film, Porco_Rosso Porco_Rosso, language, Japanese_Language	[L1: 0.73/L4: 0.84] Porco_Rosso, language, Japanese_Language Hayao_Miyazaki, film, Porco_Rosso Fantasy, titles, Porco_Rosso Porco_Rosso, written_by, Hayao_Miyazaki
(3) e_3, actor, Jonathan_Pryce (from complex FB15k)	[0.33] e_5, actor, Jonathan_Pryce	[0.00] e_3, actor, Keith_Richards	[0.00] e_3, prequel, e_4	[L1: 0.00/L4: 0.58] e_4, sequel, e_3 Keith_Richards, film, e_3 e_3, actor, Keith_Richards Action_Film, films_in_this_genre, e_3	[L1: 0.33/L4: 1.00] e_5, actor, Jonathan_Pryce Jonathan_Pryce, film, e_5 Jonathan_Pryce, film, e_4 e_3, actor, Johnny_Depp

`Japan)` ← `language(X, Japanese)` Table 5(b)). For WN18 and WN18RR, neither CP nor PT rules individually cause significant performance degradation. This observation indicates that CP and PT rules are complementary, often providing overlapping support in sparse scenarios.

These findings underscore CP rules' foundational role in semantic reasoning and PT rules' capacity to broaden explanatory scope. Their synergy achieves optimal performance. While we experimented with additional rule types—such as unary rules with dangling atoms (e.g., `country(X, Japan)` ← `language(X, Y)`)—their impact on LP explanation was negligible (<2% *δMRR* drop). This suggests that CP/PT rules uniquely balance precision and generality, whereas other rules either over-specialize (e.g., dangling atoms) or lack semantic grounding.

5.5 Case Study

We evaluate five adversarial explanation methods—Criage, Data Poisoning, KGEAttack, Kelpie, and eXpath—through three representative cases (Table 5), assessing their ability to generate minimal and interpretable explanations. For clarity, certain entities are abbreviated: e_1 refers to "Frances McDormand," e_2 to "Primetime Emmy Award for Outstanding Supporting Actress," and e_3–e_5 to films in the *Pirates of the Caribbean* series (*At World's End*, *Dead Man's Chest*, and *The Curse of the Black Pearl*).

Case 1: Path-Based Explanations Provide Intuitive Rationale. The first case examines the prediction $\langle e_1$, `award`, $e_2\rangle$. Here, KGEAttack and eXpath generate the highly effective fact $\langle e_1$, `award`, $e_2\rangle$, supported by the rule `award_nominee` ← `award′` [SC=0.815], which intuitively links the inverse relations `award_nominee` and `award`. This explanation causes a significant rank drop (head/tail ranks from 1/1 to 6/106). In contrast, Kelpie's four-fact explanation includes weaker assertions like $\langle e_2$, `award_nominee`, $X\rangle$ but lacks supporting ontological rules, making it difficult to justify. This highlights a key limitation of fact-based methods like Kelpie compared to rule-based systems such as eXpath.

Case 2: Multi-Rule Explanations Capture Comprehensive Signals. The second case involves explaining the prediction ⟨`Anime`, `country`, `Japan`⟩. KGEAttack produces a single intuitive rule: `country(X, Japan)` ← `films_in_this_genre(Anime, X)` [SC=0.846]. While this rule has high confidence, eXpath provides a more comprehensive explanation by combining four rules with $SC \geq 0.1$, including:

(1) `country(X, Japan) ← language(X, Japanese) [SC=0.669]`
(2) `country ← language, language', country [SC=0.311]`
(3) `country ← language, language', nationality [SC=0.194]`
(4) `country ← language, titles, country [SC=0.122]`

While the SC of each rule is lower than that of KGEAttack's rule, collectively, they yield a cumulative confidence greater than 0.9. This demonstrates that relying solely on one rule, as KGEAttack does, risks overlooking valuable data signals. Kelpie's explanation shares two facts with eXpath's initial rules but is heavily based on empirical signals from the embedding model and lacks the clarity and reliability of rule-based approaches.

Case 3: Multi-Fact and Tail-Related Explanations is Necessary. The third case involves the prediction $\langle e_3$, actor, Jonathan Pryce$\rangle$. Notably, eXpath (L4) delivers the most effective explanation, achieving the best attack effectiveness (δMRR = 1), while Kelpie (L4) also performs well (δMRR = 0.58). In contrast, explanations from Kelpie (L1) and other methods are largely ineffective. The consistent performance of multi-fact explanations highlights the importance of combining multiple facts, especially in dense datasets like `FB15k`, where removing a single fact often fails to impact the prediction.

Kelpie provides fact-based explanations but fails to justify the relevance of these facts in supporting the prediction. One fact, $\langle e_4, \textit{sequel}, e_3\rangle$, is supported by three high-confidence rules, including `actor ← sequel', film' [SC=0.40]`, while the remaining facts lack direct relevance. Removing this fact leaves the reverse relation $\langle e_3, \textit{prequel}, e_4\rangle$, which still supports the prediction, undermining the explanation's validity. KGEAttack also proposes a single attacking fact, $\langle e_3, \textit{prequel}, e_4\rangle$, supported by the rule `actor ← prequel, film' [SC=0.38]`. Although intuitive, this 2-hop CP rule fails for the same reason as Kelpie: the reverse relation maintains the prediction, rendering the explanation insufficient.

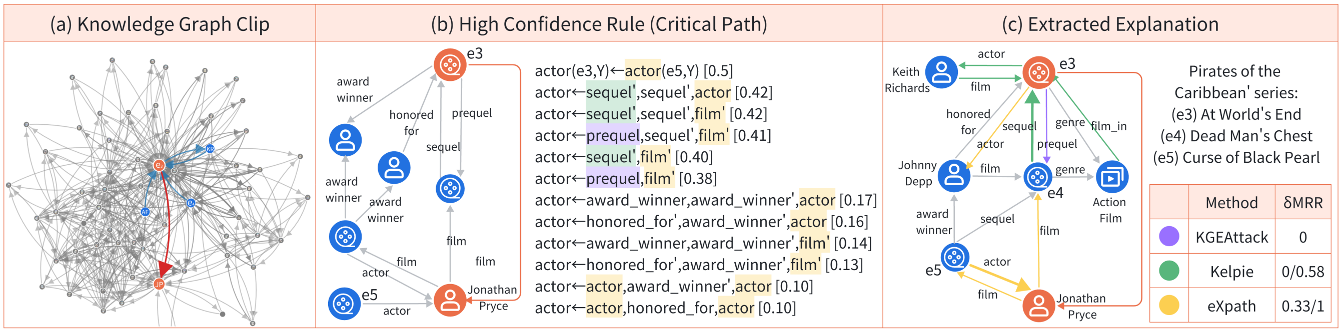

Figure 5: Explanation of the fact $\langle e_3$, actor, Jonathan_Pryce$\rangle$ predicted by LP models (ComplEx); (a) all 3-hop paths from head entity to tail entity. (b) Twelve high-confidence rules with $SC \geq 0.1$ identified by eXpath; (c) comparison of the explanation provided by KGEAttack (in purple edge), Kelpie (in green edges), and eXpath (in yellow edges).

In contrast, as shown in Figure 5(b), eXpath provides path-based explanations with supporting rules. For example, the highest-scoring fact, $\langle e_5, \textit{actor}, \textit{Jonathan\_Pryce}\rangle$, is supported by one PT and five CP rules. These rules collectively contribute to a cumulative score exceeding 0.9. Unlike KGEAttack, which focuses only on 2-hop CP rules, eXpath incorporates longer, more complex rules, capturing additional data signals. eXpath's four facts comprehensively cover all critical paths from e_3 to Jonathan Pryce, yielding a nearly perfect explanation for the prediction.

An interesting observation is that most facts selected by eXpath relate to the tail entity rather than the head entity (shown in Figure 5(c)). As depicted in Figure 5(a), the head entity (e_3) is associated with 96 triples. In contrast, the tail entity (Jonathan Pryce) is connected to only 32, making tail relations sparser and more critical for prediction. By prioritizing tail-related facts, eXpath produces more effective explanations. In contrast, Kelpie relies predominantly on head entity features, often getting trapped in local optima and missing broader contextual signals. Meanwhile, KGEAttack selects rules randomly from those it satisfies, leading to highly varied explanations and limited reliability.

User Evaluation. To assess the effectiveness, rationality, and clarity of eXpath's explanations, we conducted an in-lab user study with five graph researchers. Participants evaluated prototype diagrams comparing eXpath, KGEAttack, and Kelpie on case study examples (e.g., actor-film links in FB15k). For "effectiveness," eXpath demonstrated superior performance compared to KGEAttack and Kelpie. The majority of participants (4 out of 5) found eXpath's explanations more convincing due to the incorporation of diverse facts rather than focusing solely on head-related facts. One participant highlighted that "eXpath's utilization of analogy and co-occurrence rules resonates with how I would verify actor-film connections." In terms of "rationality," both eXpath and KGEAttack received acclaim for anchoring explanations in rules, while Kelpie's lack of rationale caused confusion. Regarding "clarity," some individuals initially found eXpath's multi-step rule-to-fact selection overwhelming ("Too many rules clutter the logic"), but this issue was alleviated by the graph visualization of rationale. While KGEAttack's simpler rules were easier to comprehend, they were deemed less informative. By leveraging comprehensive rule-based reasoning and integrating multiple facts, eXpath strikes an optimal balance between interpretability and explanatory power.

6 CONCLUSION

In this work, we introduce eXpath, a novel path-based explanation framework designed to enhance the interpretability of LP tasks on KG. By leveraging ontological closed path rules, eXpath provides semantically rich explanations that address challenges such as scalability and relevancy of path evaluation on embedding-based KGLP models. Extensive experiments on benchmark datasets and mainstream KG models demonstrate that eXpath outperforms the best existing method by 12.4% on δMRR in terms of the most important multi-fact explanations. A higher improvement of 20.2% is achieved when eXpath is further combined with existing methods. Ablation studies validate that the CP rule in our framework plays a central role in the explanation quality, with its removal leading to a 13.5%~27% average drop in performance. These findings suggest that ontological rules, such as CP and PT rules, are not only interpretable but also essential for bridging the gap between symbolic reasoning and subsymbolic embeddings.

Future work will focus on developing interactive visualization tools to enhance the accessibility and interpretability of eXpath's path-based explanations. These tools will allow users to explore critical paths and ontological rules supporting each prediction. Building on this, we plan to conduct a user study involving domain experts and data scientists to quantify the alignment of path-based explanations with human reasoning and assess their effectiveness in improving trust and transparency in KG predictions.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China (2021YFB3500700), NSFC Grant 62172026, National Social Science Fund of China 22&ZD153, the Fundamental Research Funds for the Central Universities, State Key Laboratory of Complex & Critical Software Environment (SKLCCSE). Yongxin Tong's research is supported by National Science Foundation of China (NSFC) (Grant Nos. 62425202, 62336003).

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *International Semantic Web Conference (ISWC)*. Springer, 722–735.

[2] Patrick Betz, Christian Meilicke, and Heiner Stuckenschmidt. 2022. Adversarial explanations for knowledge graph embeddings. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 2022. 2820–2826.

[3] Peru Bhardwaj, John Kelleher, Luca Costabello, and Declan O'Sullivan. 2021. Adversarial attacks on knowledge graph embeddings via instance attribution methods. *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2021).

[4] Rajarshi Bhowmik and Gerard de Melo. 2020. Explainable link prediction for emerging entities in knowledge graphs. In *International Semantic Web Conference (ISWC)*. Springer, 39–55.

[5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*. 1247–1250.

[6] Heng Chang, Jiangnan Ye, Alejo Lopez-Avila, Jinhua Du, and Jia Li. 2024. Path-based Explanation for Knowledge Graph Completion. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 231–242.

[7] Yang Chen, Daisy Zhe Wang, and Sean Goldberg. 2016. ScaLeKB: scalable learning and inference over large knowledge bases. *VLDB Journal* 25 (2016), 893–918.

[8] Yashrajsinh Chudasama. 2023. Exploiting Semantics for Explaining Link Prediction Over Knowledge Graphs. In *European Semantic Web Conference (ESWC)*. Springer, 321–330.

[9] U.S. Congress. 2021. Artificial Intelligence Accountability Act of 2021. Available at: https://www.congress.gov/bill/117th-congress/house-bill/3463.

[10] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 32.

[11] Wenfei Fan, Wenzhi Fu, Ruochun Jin, Ping Lu, and Chao Tian. 2022. Discovering association rules from big graphs. *Proceedings of the VLDB Endowment (PVLDB)* 15, 7 (2022), 1479–1492.

[12] Wenfei Fan, Xin Wang, Yinghui Wu, and Jingbo Xu. 2015. Association rules with graph patterns. *Proceedings of the VLDB Endowment (PVLDB)* 8, 12 (2015), 1502–1513.

[13] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. 2015. Fast rule mining in ontological knowledge bases with AMIE+. *VLDB Journal* 24, 6 (2015), 707–730.

[14] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*. 413–422.

[15] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2018. Knowledge graph embedding with iterative guidance from soft rules. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 32.

[16] Adrianna Janik and Luca Costabello. 2022. Explaining Link Predictions in Knowledge Graph Embedding Models with Influential Examples. *arXiv preprint arXiv:2212.02651* (2022).

[17] Niraj Kumar-Singh, Gustavo Polleti, Saee Paliwal, and Rachel Hodos-Nkhereanye. 2024. LinkLogic: A New Method and Benchmark for Explainable Knowledge Graph Predictions. *arXiv preprint arXiv:2406.00855* (2024).

[18] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning (ICML)*. 2863–2872.

[19] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NeurIPS)* 30 (2017).

[20] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), 19620–19631.

[21] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. 2014. Yago3: A knowledge base from multilingual wikipedias. In *7th Biennial Conference on Innovative Data Systems Research (CIDR)*.

[22] Christian Meilicke, Patrick Betz, and Heiner Stuckenschmidt. 2021. Why a naive way to combine symbolic and latent knowledge base completion works surprisingly well. In *3rd Conference on Automated Knowledge Base Construction (AKBC)*.

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

[24] Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2020. Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*. 3137–3143.

[25] MK Nallakaruppan, Himakshi Chaturvedi, Veena Grover, Balamurugan Balusamy, Praveen Jaraut, Jitendra Bahadur, VP Meena, and Ibrahim A Hameed. 2024. Credit Risk Assessment and Financial Decision Support Using Explainable Artificial Intelligence. *Risks* 12, 10 (2024), 164.

[26] Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. 2018. Scalable Rule Learning via Learning Representation. In *International Joint Conference on Artificial Intelligence (IJCAI)*. 2149–2155.

[27] Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. 2019. An embedding-based approach to rule learning in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 33, 4 (2019), 1348–1359.

[28] Pouya Pezeshkpour, CA Irvine, Yifan Tian, and Sameer Singh. 2019. Investigating Robustness and Interpretability of Link Prediction via Adversarial Modifications. In *Proceedings of NAACL-HLT*. 3336–3347.

[29] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1135–1144.

[30] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 32.

[31] Andrea Rossi, Donatella Firmani, Paolo Merialdo, and Tommaso Teofili. 2022. Explaining link prediction systems based on knowledge graph embeddings. In *Proceedings of the International Conference on Management of Data (SIGMOD)*. 2062–2075.

[32] Andrea Rossi, Donatella Firmani, Paolo Merialdo, and Tommaso Teofili. 2022. Kelpie: an explainability framework for embedding-based link prediction models. *Proceedings of the VLDB Endowment (PVLDB)* 15, 12 (2022), 3566–3569.

[33] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*. PMLR, 2071–2080.

[34] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 28.

[35] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in Neural Information Processing Systems (NeurIPS)* 32 (2019).

[36] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. In *International Conference on Machine Learning (ICML)*. PMLR, 12241–12252.

[37] Hengtong Zhang, Tianhang Zheng, Jing Gao, Chenglin Miao, Lu Su, Yaliang Li, and Kui Ren. 2019. Data poisoning attack against knowledge graph embedding. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (2019).

[38] Shichang Zhang, Yozen Liu, Neil Shah, and Yizhou Sun. 2022. Gstarx: Explaining graph neural networks with structure-aware cooperative games. *Advances in Neural Information Processing Systems (NeurIPS)* 35 (2022).

[39] Shichang Zhang, Jiani Zhang, Xiang Song, Soji Adeshina, Da Zheng, Christos Faloutsos, and Yizhou Sun. 2023. PaGE-Link: Path-based graph neural network explanation for heterogeneous link prediction. In *Proceedings of the ACM Web Conference (WWW)*. 3784–3793.

[40] Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019. Iteratively learning embeddings and rules for knowledge graph reasoning. In *The World Wide Web Conference (WWW)*. 2366–2377.

[41] Dong Zhao, Guojia Wan, Yibing Zhan, Zengmao Wang, Liang Ding, Zhigao Zheng, and Bo Du. 2023. KE-X: Towards subgraph explanations of knowledge graph embedding based on knowledge information gain. *Knowledge-Based Systems* 278 (2023), 110772.