Machine Learning (2026) 115:136
https://doi.org/10.1007/s10994-026-07052-8

From Latent to Lucid: Transforming Knowledge Graph Embeddings into Interpretable Structures with KGEPrisma

Christoph Wehner[1,2] · **Chrysa Iliopoulou**[1] · **Ute Schmid**[2] · **Tarek R. Besold**[1]

Received: 7 May 2025 / Revised: 28 January 2026 / Accepted: 8 April 2026

Abstract

In this paper, we introduce a post-hoc and local explainable AI method tailored for Knowledge Graph Embedding (KGE) models. These models are essential to Knowledge Graph Completion yet criticized for their black-box nature. Despite their success in capturing the semantics of knowledge graphs through high-dimensional latent representations, their inherent complexity poses substantial challenges to explainability. While existing methods like Kelpie use resource-intensive perturbation to explain KGE models, our approach directly decodes the latent representations encoded by KGE models, leveraging the smoothness of the embeddings, which follows the principle that similar embeddings reflect similar behaviours within the Knowledge Graph, meaning that nodes are similarly embedded because their graph neighbourhood looks similar. This principle is commonly referred to as smoothness. By identifying symbolic structures, in the form of triples, within the subgraph neighborhoods of similarly embedded entities, our method identifies the statistical regularities on which the models rely and translates these insights into human-understandable symbolic rules and facts. This bridges the gap between the abstract representations of KGE models and their predictive outputs, providing clear and interpretable insights. The contributions include a novel post-hoc and local explainable AI method for KGE models, which provides immediate and faithful explanations without retraining, thereby facilitating real-time application on large-scale knowledge graphs. The method's flexibility enables the generation of rule-based, instance-based, and analogy-based explanations, meeting diverse user needs. Extensive evaluations show the effectiveness of our approach in delivering faithful and well-localized explanations, enhancing the transparency and trustworthiness of KGE models.

Keywords Knowledge graphs · Knowledge graph embedding · Explainability · XAI · Interpretability

Editor: Steven Schockaert.

Extended author information available on the last page of the article

Published online: 25 May 2026

1 Introduction

Knowledge Graphs (KG), despite their vast potential for structuring and leveraging information, are notoriously incomplete (Ji et al., 2022; Eirich et al., 2023; Bahr et al., 2025). To mitigate this issue, link prediction has emerged as a technique for uncovering previously unknown links within these graphs (Hogan et al., 2021). Knowledge Graph Embedding (KGE) models have become the de facto standard due to their ability to capture the complex relationships and semantics embedded within the graph structure through high-dimensional latent representations (Ji et al., 2022). However, these models are criticized for their black-box nature (Schramm et al., 2023), which obscures the underlying mechanisms and rationales behind their predictions (Schwalbe & Finzel, 2023), posing challenges for explainability in critical applications (Wehner et al., 2022, 2023).

Explainable Artificial Intelligence (XAI) has made significant progress in making the opaque decision-making processes of complex black-box models more transparent (Schwalbe & Finzel, 2023). Despite the development of explainable methods such as LIME (Ribeiro et al., 2016), SHAP (Lundberg et al., 2017), Layer-wise Relevance Propagation (Montavon et al., 2019), and Integrated Gradients (Sundararajan et al., 2017), applying these methods to KGE models presents a non-trivial challenge. These methods traditionally work by attributing parts of the input as relevant or not to the model's output. However, embedding-based link prediction operates differently. It relies on the latent representations of entities and relations in a triple (head, relation, tail) as input to an interaction function to compute a score. This score is then used to create an ordinal ranking of the plausibility of different permutations for the head, relation, or tail (Ji et al., 2022). In this context, simply assigning relevance to the latent representations of the triple provides minimal insight into the underlying rationale of the prediction. The inherent complexity of these embeddings and the abstract nature of the relations they capture make it difficult to draw clear, interpretable connections between input features and the model's output.

This work presents the XAI method KGEPrisma. KGEPrisma leverages the principle that KGE models encode a KG's statistical regularities into latent representations, reflecting the KG's structure and interactions. Central to the method is the smoothness of embeddings, meaning that entities with similar embeddings behave similarly within the KG (Bengio et al., 2013). KGEPrisma decodes these embeddings by identifying distinct symbolic structures in the subgraph neighborhoods of entities with similar embeddings, revealing the model's relied-upon symbolic regularities. These structures can be represented as human-understandable symbolic rules and facts, clarifying the predictive patterns in localized subgraphs (cf. Fig. 1). Our evaluations show that the proposed method outperforms state-of-the-art methods regarding faithfulness to the KGE model's decision process and that the explainable evidence is better centered around a region of interest.

This work contributes (1) a novel, local and post-hoc explainable AI method for KGEs. In contrast to others, our method is aligned with the operational mechanics of KGE models, ensuring explanations are faithful to the model's decision-making process, localized around a region of interest and immediate, thereby eliminating the need to retrain the model on occluded training data. This enables real-time, scalable explanations within extensive KGs. (2) Furthermore, our method is versatile, producing explanations in various forms, including rule-based, instance-based, and analogy-based, making it adaptable to diverse user requirements. (3) Through comprehensive evaluations, we demonstrate that our approach

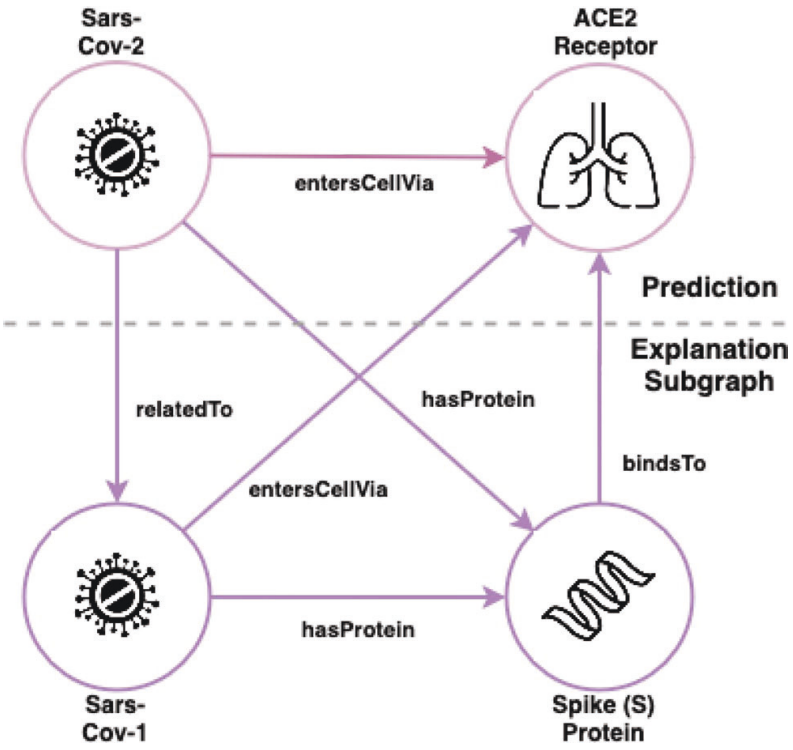

Fig. 1 KGEPrisma generates explanations for KGE models in the form of subgraphs, uncovering the reasoning of the KGE model and building trust in the model's prediction. For example, the KGE model predicts that Sars-Cov-2 enters the respiratory cell via the ACE2 receptor. The explanation subgraph by KGEPrisma uncover that Sars-Cov-2 has a Spike (S) protein. Furthermore, it shows that Sars-Cov-2 is related to Sars-Cov-1; thus, both are likely to behave similar. Sars-Cov-1 also has a Spike (S) protein, which Sars-Cov-1 uses to bind to the ACE2 receptor, enabling it to enter the respiratory cell

performs well compared to existing state-of-the-art methods regarding faithfulness to the model's decision-making process and providing more relevant explanations centered on the user's region of interest.

2 Preliminaries

This section briefly introduces Knowledge Graphs, Knowledge Graph Completion (KGC) and Knowledge Graph Embeddings and fixes some notations to be used later.

2.1 Knowledge Graph

A Knowledge Graph is a directed labeled graph G (Hogan et al., 2021), consisting of triples (i.e., facts) $G \subseteq E \times R \times E$ from the entity set $e \in E$ and relation set $r \in R$, allowing the traversal of a triple (e^{head}, r, e^{tail}) from a head to a tail entity via a relation. Triples can be expressed as grounded binary predicates $r(e^{head}, e^{tail})$. The relation acts as the binary predicate and the entities as the grounding constants. A KG assigns each entity and relation a symbolic label (e.g., name). KGs are structured according to a semantic schema $s : E \rightarrow C$ (Hogan et al., 2021). This schema categorizes entities into classes C within the KG's domain, facilitating the storage and retrieval of semantically rich, relational data.

Nonetheless, the construction of KGs demands substantial expert knowledge, leading to the common issue of incomplete knowledge graphs.

2.2 Knowledge Graph Completion

Knowledge Graph Completion (i.e., Link Prediction) addresses the challenge of inherently incomplete KGs (Hogan et al., 2021). For KGs, there exists a subset of correct but unknown triples $G_{unkown} \subseteq E \times R \times E$ that do not intersect with the existing graph G. KGC aims to uncover these missing facts by exploiting the regularities and patterns inherent in the KG, thus deducing the unknown triples. In practice, KGC models are queried with partial triples $(e^{head}, r, ?)$, $(?, r, e^{tail})$, or $(e^{head}, ?, e^{tail})$, seeking to complete these by predicting the missing entity or relation. The model then generates a ranked list of candidates. The higher the rank, the more plausible it is for a candidate to complete the triple (Rossi et al., 2021).

2.3 Knowledge Graph Embeddings

Knowledge Graph Embedding models enable KGC by learning latent space representations $v \in \mathbb{R}^n$ (i.e., embeddings) for entities and relations within a knowledge graph (Ali et al., 2021), where n is the number of embedding dimensions. An interaction function i assigns a score to the embedding of a triple.

$$i : E \times R \times E \rightarrow \mathbb{R} \tag{1}$$

The score allows the creation of an ordinal ranking. A higher rank indicates a greater plausibility of the triple being true. This scoring mechanism is crucial for optimizing the embeddings to favor existing triples over corrupted ones, ensuring that the embeddings reflect the KG's symbolic regularities. Consequently, embeddings are smooth, meaning that entities exhibiting similar behavior within the graph are represented by similar embeddings (Bengio et al., 2013; Rossi et al., 2021; Ji et al., 2022). Models such as TransE (Bordes et al., 2013) optimize embeddings by aligning the sum of entity and relation embeddings with the missing entity's embedding. DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016) further refine this approach by implementing a trilinear dot product and extending capabilities to capture non-symmetric relationships. Other models like ConvE (Dettmers et al., 2018) utilize convolutions in the interaction function. Despite the advancements in KGE models, the complexity and abstractness of the embeddings pose significant challenges in establishing clear, interpretable links between input features and model outputs.

3 Method

The approach is rooted in the understanding that KGE models encapsulate the symbolic patterns of a KG in smooth latent representations, encoding the graph's topology and the interactions between its entities (Bengio et al., 2013). Thus, entities sharing similar embeddings exhibit comparable behavior within the symbolic structure of the KG. By analyzing the subgraph neighborhoods of these similar entities, symbolic regularities are discovered, in

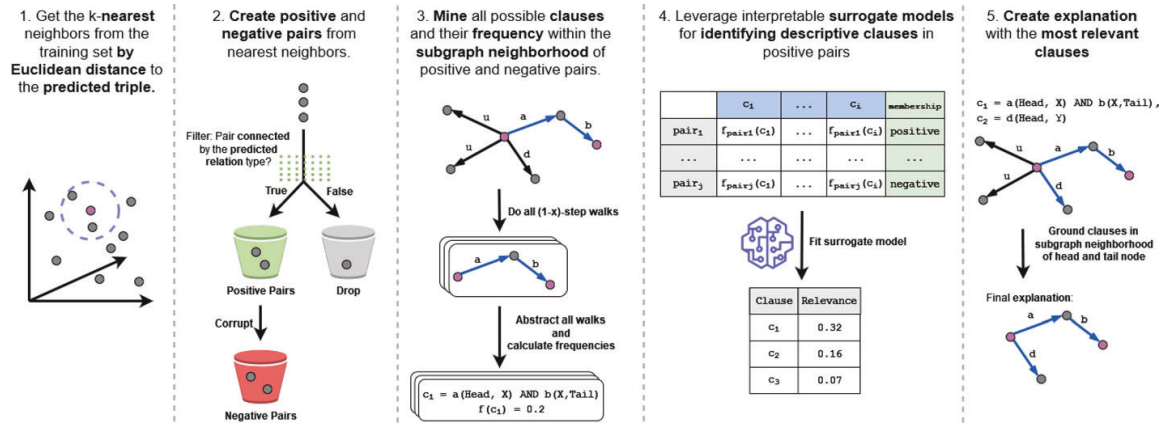

Fig. 2 KGEPrisma generates explanations of KGE models in five steps. The five steps are discussed in detail in Sect. 3

the form of conjunctive clauses[1], that KGE models depend on. KGEPrisma translates these regularities into symbolic rules, or triples comprehensible to humans, thereby uncovering the rationale behind the models' predictions in local subgraph contexts (cf. Fig. 1). This allows KGEPrisma to post hoc and locally explain the predicted triple $(e_p^{head}, r_p, e_p^{tail})$, by accessing the training knowledge graph and the embeddings learned by the KGE model.

The method is build on five steps (cf. Fig. 2):

1. Get k-nearest neighbors in the latent embedding space to the predicted triple (cf. Fig. 2 Step 1),
2. Create positive and negative entity-pairs from the nearest neighbors (cf. Fig. 2 Step 2),
3. Mine all possible clauses and their frequency within the subgraph neighbourhood of the pairs (cf. Fig. 2 Step 3),
4. Identify the most descriptive clauses for positive entity-pairs with the help of a surrogate model (cf. Fig. 2 Step 4), and
5. Ground the most descriptive clauses to create an explanation (cf. Fig. 2 Step 5).

In the following section, KGEPrisma is introduced step by step.

3.1 Step 1: Identifying K-Nearest Neighbors

In the initial step of the post hoc explainability method, the embedding $v_{predicted}$ of a given predicted triple is taken as input. The creation of triple embeddings is KGE model-dependent. Appendix A provides examples of KGE models and their methods for obtaining triple embeddings. In its simplest form, $v_{predicted} = [v^{head}; v^r; v^{tail}]$ is a concatenation of the entity and relation embeddings.

The k-nearest neighbor embeddings $v_1, v_2, ..., v_k$ are then retrieved from the set of all KGE model training triple embeddings V_{train}, based on the Euclidean distance (cf. Fig. 2 Step 1). The Euclidean distance is used as it demonstrates robust performance in finding

[1] Clauses are commonly referred to in their disjunctive form, which means that the clause is true whenever at least one of the letters of that form is true. In this paper, clauses are referred to in their conjunctive form. This means that a clause is true when all of the literals that form it are true.

similarity-based explanations in previous work conducted on image data (Hanawa et al., 2021).[2] The retrieval is described by the equation:

$$kNN(v_{predicted}) = \underset{v \in V_{train}}{\text{argmin}}_{\,k} ||v_{predicted} - v||_2 \tag{2}$$

In this equation, $\text{argmin}_{\,k}$ identifies the k embeddings v that yield the smallest Euclidean distances to $v_{predicted}$, thus isolating the embeddings in the latent space that are most likely to exhibit symbolic regularities in common with the predicted triple. Thus, KGEPrisma considers the local decision surface of the KGE model to explain a triple, enabling an efficient computation and coupling the explanation closely to the model behaviour. This step guarantees that the explanation generated in downstream steps reflects the internal mechanics of the KGE model by localizing the explanation around the training instances that the model learned to see and treat similarly. The embeddings are then mapped back to their symbolic triple representations, the relationship symbol is dropped and the entity-pairs are stored in $N = ((e_1^{head}, e_1^{tail}), (e_2^{head}, e_2^{tail}), ..., (e_k^{head}, e_k^{tail}))$. In the next step, positive and negative pairs are created with the help of the pairs in N.

3.2 Step 2: Create Positive and Negative Entity-Pairs

Step two constructs positive and negative entity pairs (cf. Fig. 2 Step 2). A nearest neighbor pair $(e_i, e_j) \in N$ belongs to the positive set P^+ if (e_i, r_p, e_j) is a fact in G, where r_p is the relation type of the predicted triple we want to explain. Conversely, a pair (e_k, e_l) is in the negative set P^- if (e_k, r_p, e_l) does not exist in G, representing a corrupted version of a positive pair.

$$\begin{aligned} P^+ &= \{(e_i, e_j) \in N | (e_i, r_p, e_j) \in G\} \\ P^- &= \{(e_k, e_l) | (e_k = e_i \veebar e_l = e_j) \\ &\qquad \wedge (e_k, r_p, e_l) \notin G\}, \\ &\qquad with\ (e_i, e_j) \in P^+ \wedge e \in E \end{aligned} \tag{3}$$

The process results in two sets, P^+ containing pairs that are connected by the predicted relation type and have a similar latent representation to the predicted triple, and P^-, which includes pairs that serve as corrupted versions of the positive pairs. It is important to emphasize that P^+ holds the triples from which the KGE model learned its behavior. In practice, one corrupted pair for every positive pair in P^+ is sampled by randomly switching a positive pair's head or tail entity with a random entity from the KG. This procedure is similar to the stochastic local closed-world assumption applied while training KGE models (Ali et al., 2021).

[2] The cosine distance was used in an ablation study; however, no meaningful impact on the method was observed.

3.3 Step 3: Mining Clauses Frequency

In the third step, the method abstracts the subgraph neighborhoods of entity pairs into conjunctive clauses and computes their frequencies (cf. Fig. 2 Step 3). This process aims to identify symbolic regularities, in the form of clauses, that the KGE model implicitly relies on for its predictions.

Algorithm 1 Clause Mining and Frequency Calculation

Input: Positive pairs P^+, negative pairs P^-, knowledge graph G
Parameter: Maximum walk length x
Output: Dictionary D mapping each pair to its unique clauses and frequencies
for each pair $(e_{\text{head}}, e_{\text{tail}}) \in P^+ \cup P^-$ **do**
    Initialize an empty multiset $S_{(e_{\text{head}}, e_{\text{tail}})}$ to store clauses
    for each walk w in G of length 1 to x, starting or ending at e_{head} or e_{tail}, without including them as intermediate nodes **do**
        Apply schema mapping $s : E \to C \cup \{\text{Head}, \text{Tail}\}$ to abstract entities in w
        Obtain clause c from the abstracted walk
        Add c to $S_{(e_{\text{head}}, e_{\text{tail}})}$
    end for
    for each unique clause c in $S_{(e_{\text{head}}, e_{\text{tail}})}$ **do**
        Compute frequency $f_c = \frac{|\{c \in S_{(e_{\text{head}}, e_{\text{tail}})}\}|}{|S_{(e_{\text{head}}, e_{\text{tail}})}|}$
        Store (c, f_c) in $D_{(e_{\text{head}}, e_{\text{tail}})}$
    end for
end for

Algorithm 1 Clause mining and frequency calculation

For each entity pair $(e_{\text{head}}, e_{\text{tail}})$ in the combined set $P = P^+ \cup P^-$, walks of length from 1 to x are constructed in the knowledge graph G. These walks start or end at either e_{head} or e_{tail}, but do not include them as intermediate nodes. Each walk w represents a sequence of entities and relations anchored in e_{head} or e_{tail}.

To abstract the walks into clauses, a schema mapping $s : E \to C \cup \{\text{Head}, \text{Tail}\}$ is applied, where E is the set of entities, C is the set of classes, and Head and Tail are special classes assigned to e_{head} and e_{tail}, respectively. This mapping replaces each entity in the walk with its class, creating an abstract representation of the walk that preserves structural patterns while generalizing over specific entities.

Each abstracted walk corresponds to a conjunctive clause c, which can be thought of as a logical expression capturing the relationships between classes in the neighborhood of the entity pair. By including the special classes Head and Tail, the abstraction maintains anchors to the original entities, enabling later grounding.

The method thus captures following conjunctive clause types:

$$r_1(A, W) \wedge r_2(X, Y) \wedge ... \wedge r_m(Y, B) \tag{4}$$

$$r_1(A, W) \wedge r_2(X, Y) \wedge ... \wedge r_m(Y, Z) \tag{5}$$

$$r_1(X, Y) \wedge r_2(Y, Z) \wedge ... \wedge r_m(Z, A), \tag{6}$$

where $A, B \in \{Head, Tail\}$, the variable $m \leq x$ is the actual step length, and W, X, Y, $Z \in C$ are classes.

For each unique clause c obtained from the walks associated with an entity pair, its frequency f_c is calculated. The frequency is the proportion of times the clause appears in the set of all clauses $S_{(e_{\text{head}}, e_{\text{tail}})}$ for that pair. This frequency reflects the relative significance of the clause in the subgraph neighborhood of the entity pair.

The tuple (c, f_c) is stored in the dictionary $D_{(e_{\text{head}}, e_{\text{tail}})}$ for each pair, resulting in a mapping of each entity pair to its unique clauses and their frequencies. Algorithm 1 details this procedure.

This process ensures that the frequencies of clauses are mapped for each entity pair, which are then used in the following step to identify the symbolic regularities that the KGE model relies on for its predictions.

3.4 Step 4: Leveraging Surrogate Models for Identifying Descriptive Clauses in Positive Entity Pairings

The dictionary D establishes a classic tabular machine learning setup, wherein instances are represented by entity pairs, features by conjunctive clauses, and values by the frequency of the clauses. The labels are categorized as positive or negative based on the entity pair's membership of P^+ or P^- (cf. Fig. 2 Step 4). The objective is to identify which feature (clause) contributes the most to classifying an entity pair as positive. This is achieved by utilizing surrogate models, which allows extracting the feature importances to interpret the complex relationships within the data. A natural question arises: why not use clause frequencies directly to identify the most relevant clauses? Using frequency alone does not adequately capture the complex relationship between clauses and classification outcomes. For instance, a high clause frequency does not necessarily imply that a pair should be classified as positive. In some cases, the mere presence of a certain clause is sufficient for a positive classification, while in others, the absence (frequency of zero) of a clause is the deciding factor. Surrogate models are able to capture these complex, sometimes non-linear, relationships between clause frequencies and the classification outcome. This approach is inspired by established XAI methods such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg et al., 2017), which similarly employ surrogate models to approximate decision boundaries of black-box models.

The goal is thus to assign each clause a score by which it is ranked in accordance with its relevance for classifying an entity pair as positive or negative.

For KGEPrisma, mean decrease in impurity (Nembrini et al., 2018), K-Lasso (Ribeiro et al., 2016), and HSIC-Lasso (Yamada et al., 2014) are studied and compared to identify feature importance.

The **Mean Decrease in Impurity** (MDI) quantifies each clause's role in classifying positive or negative samples in *D* through an ensemble of decision trees (Nembrini et al., 2018). This process entails iterative data splitting based on clauses that maximize impurity reduction, employing the Gini impurity (Nembrini et al., 2018) as a measure of this reduction. The Gini impurity for a dataset $d \subseteq D$ is defined as:

$$Gini(d) = 1 - \sum_i p(i|t)^2 \tag{7}$$

where *p*(*i*|*t*) denotes the proportion of class $i \in \{positive, negative\}$ at tree-node (dataset) $d \subseteq D$, adjusted by weights α reflecting the Euclidean distance of a pair embedding v_i to the predicted pairs's embedding v_p. The weight is defined as an exponential kernel $\alpha(v_i) = \exp(-||(v_p, v_i)||_2^2/\sigma^2)$ with kernel width σ. This assigns a higher impact to pairs that are perceived by the model as similar. The Gini impurity evaluates the likelihood of mislabeling an element if randomly assigned based on the subset's label distribution, serving as a statistical regularity indicator in *D*. Most relevant features reduce the Gini impurity of a dataset by the most over all nodes within the tree.

The impurity reduction ($\Delta Gini$) (Nembrini et al., 2018) from splitting at tree-node *d* on clause *c*, yielding "positive" (*L*) and "negative" (*R*) child nodes, is given by:

$$\Delta Gini(d, c) = Gini(d) - (\frac{N_L}{N_d} Gini(L) + \frac{N_R}{N_d} Gini(R)) \tag{8}$$

here, N_d, N_L, and N_R represent the weighted counts of samples at the parent node and in each child node, respectively.

The MDI for a clause across the ensemble is the impurity reductions' mean, weighted by the samples reaching the nodes where the feature splits the data:

$$MDI(c) = 1 - \frac{\gamma_c}{N} \sum_{d_c \in D} \Delta Gini(d_c, c) N_d \tag{9}$$

where d_c is a node split on clause *c*, and N_d is the total sample count in d_c. A weighted frequency co-factor γ is applied to the MDI of a clause. It is defined as:

$$\gamma(c) = \begin{cases} 1 & \text{if } \sum_{f_c \in D_+} f_c \geq \sum_{f_c \in D_-} f \\ -1 & \text{otherwise} \end{cases} \tag{10}$$

This allows to weight in if a clause is more frequent in positive instances D_+ or negative instances D_+ of the dictionary.

MDI thereby assesses clause importance, identifying those crucial for positive pair classifications within *D*, revealing key statistical patterns in similar sub-graph neighborhoods.

Nonetheless, MDI may favor features with higher cardinality, such as those capturing multi-hop regularities, over binary property relations, due to inherent biases of MDI toward features with broader variation (Nembrini et al., 2018).

The **K-Lasso** method uses a linear model, specifically ridge regression (Hoerl & Kennard, 2000), to weight each clause contribution in the classification task within the dictionary D. The method learns a weight for every feature (clause), employing linear least squares with Euclidean regularization to optimize the model (Ribeiro et al., 2016). The objective function minimized by this model is formalized as:

$$min(\sum_{(e_i,e_j)\in D} \alpha_{(e_i,e_j)}(y_{(e_i,e_j)} - C^T_{(e_i,e_j)}w) + \beta||w||_2^2) \tag{11}$$

here, $y \in \{1, -1\}$ is the label (positive, negative) of the entity pair $(e_i, e_j) \in D$, C is the feature vector holding the frequencies of all clauses of an entity pair, and w is the vector of weights corresponding to the clauses. The kernel $\alpha(v_{(e_i,e_j)}) = \exp(||(v_p, v_{(e_i,e_j)})||_2^2/\sigma^2)$ with kernel width σ scales the impact of pairs that are perceived by the model as similar, allowing for differential emphasis on instances that are perceived by the model as closer to the predicted triple. The parameter β is for the Euclidean regularization, penalizing the sum of squared weights to prevent overfitting. After fitting the surrogate model, the learned weights w for each feature provide a direct measure of feature importance. These weights reflect the contribution of each clause to the prediction task, with larger absolute values indicating greater importance. This enables the identification of the most relevant clauses that contribute to classifying entity pairs as positive or negative, providing insights on the underlying statistical regularities captured by the KGE models.

Compared to MDI, K-Lasso is not biased towards features with high cardinality. However, it permits only linear feature selection, which may not capture complex relationships between features in certain datasets effectively (Yamada et al., 2014).

The **Hilbert-Schmidt Independence Criterion Lasso** (HSIC-Lasso) is a supervised nonlinear feature selection methodology aimed at identifying a subset of input features relevant to predicting output values. As an extension of the standard Lasso, HSIC-Lasso incorporates a feature-wise kernelized Lasso to capture nonlinear dependencies between inputs and outputs. This enables it to identify non-redundant features with a significant statistical dependence on the output values (Yamada et al., 2014; Huang et al., 2023).

The optimization problem of HSIC-Lasso is formalized as follows (Yamada et al., 2014):

$$\min_{w_1,\ldots,w_d} \frac{1}{2}||\mathrm{L} - \sum_k w_k \tilde{\mathbf{K}}^{(k)}||_F^2 + \lambda \sum_k |w_k| \quad , with \quad w_1, \ldots, w_d \geq 0 \tag{12}$$

where $|\cdot|_F$ denotes the Frobenius norm, $\tilde{\mathbf{K}}^{(k)}$ represents the centered Gram matrix for the k-th feature, and L is the centered Gram matrix for the output y.

After training the surrogate model, coefficients (w) are obtained, which, when multiplied by the frequency co-factor γ (cf. Eq. 10), identify clauses predominantly associated with the sub-graph neighborhood of positive entity pairs. The coefficients reflect how relevant each clause is to the prediction.

3.5 Step 5: Generating Explanations from the Most Descriptive Clauses

After obtaining the most descriptive clauses from the surrogate model, they are used to generate explanations of the KGE model's prediction (cf. Fig. 2 Step 5). The approach allows generating three explanation types, each catering to different aspects of user needs: rule-based, instance-based, and analogy-based. It is important to note that all explanation types are based on the same clauses. The difference is in their grounding triples, which gives the user different perspectives on why the model believes the predicted triple is true and thus increases the trust in the prediction. Table 1 provides an example of that.

Rule-based explanations are derived by appending an implication to the most relevant clauses, thus forming a set of symbolic rules. These rules express the symbolic regularities that the KGE model has learned to predict a missing link. For instance, if the most descriptive clause extracted for the predicted triple (*Alice*, *knows*, *Bob*) is $knows(Head, Person) \wedge works\_with(Person, Tail)$, the rule is formulated as $knows(Alice, Person) \wedge works\_with(Person, Bob) \rightarrow knows(Alice, Bob)$. This implies that if *Alice knows* someone of the class *Person*, and this *Person* $works\_with$ *Bob*, the triple (*Alice*, *knows*, *Bob*) is predicted. Rule-based explanations provide the user with a broader justification of why the predicted triple is believed to be true by showing a pattern that is dominant in the subgraphs around similar embedded training triple instances.

Instance-based explanations are generated by grounding the most relevant clauses in the knowledge graph. Grounding, in logical terms, means replacing the variables in a clause with specific constants from the domain, thus instantiating the clause. For example, if $knows(Head, Person) \wedge works\_with(Person, Tail)$ is a clause and the predicted triple is (*Alice*, *knows*, *Bob*), the grounding would be $knows(Alice, Tom) \wedge works\_with(Tom, Bob)$. This means that *Alice knows Tom*, and *Tom* $works\_with$ *Bob*; thus, *Alice knows Bob*. This type of explanation provides the concrete triples that led the model to predict the *knows*-relation between *Alice* and *Bob*.

Analogy-based explanations focus on how the model behaves in similar situations by grounding the literals with the head and tail of the pair from P^+ that is closest in terms of Euclidean distance to the predicted pair. This approach demonstrates the model's behavior on similar instances, for which the prediction is confirmed to be true by the training facts. For example, if the nearest pair to (*Alice*, *knows*, *Bob*) in P^+ is (*Carol*, *Dave*), and $knows(Head, Person) \wedge works\_with(Person, Tail)$ is a clause, the grounding would

Table 1 Comparison of explanation types generated from examplatory most descriptive clauses, given the predicted triple (*Alice*, *knows*, *Bob*) and its closest positive neighbour (*Carol*, *Dave*)

	1st Clause	2nd Clause	..
Clause	*knows*(*Head*, *Person*) $\wedge\ works\_with(Person, Tail)$	*knows*(*Head*, *Person*) $\wedge\ sibling\_of(Person, Tail)$	..
Relevance	0.54	0.31	..
Rule-based explanation	*knows*(*Alice*, *Person*) $\wedge\ works\_with(Person, Bob)$ $\rightarrow\ knows(Alice, Bob)$	*knows*(*Alice*, *Person*) $\wedge\ sibling\_of(Person, Bob)$ $\rightarrow\ knows(Alice, Bob)$	..
Instance-based explanation	*knows*(*Alice*, *Tom*) $\wedge\ works\_with(Tom, Bob)$	*knows*(*Alice*, *Pedro*) $\wedge\ sibling\_of(Pedro, Bob)$	..
Analogy-based explanation	*knows*(*Carol*, *Anja*) $\wedge\ works\_with(Anja, Dave)$	*knows*(*Carol*, *Jan*) $\wedge\ sibling\_of(Jan, Dave)$	..

For simplicity, the table shows only the two most relevant clauses identified by KGEPrisma

be $knows(Carol, Anja) \wedge works\_with(Anja, Dave)$. This means that *Alice knows Tom* because first of all *Carol* and *Dave* are similar to *Alice* and *Tom*. And for *Carol* and *Dave*, it is a fact that they *know* each other because *Coral knows Anja*, and *Anja* $works\_with$ *Dave*. Thus, because this pattern also works by analogy for *Alice* and *Tom*, *Alice* has a high chance of knowing *Tom*. This shows an analogous situation where the model applied similar decision-making.

The resultant triples are then presented to the user. This allows the user to uncover the hidden symbolic regularities that the KGE model has learned and utilized to predict the missing link. Such explanations not only enhance the transparency of the model but also increase the user's trust by making the model's predictions interpretable and verifiable.

4 Evaluation

The evaluation section outlines the protocol to assess the faithfulness (Hedström et al., 2023) of KGEPrisma to KGE model behavior, comparing it against a retraining baseline, a local random baseline, a global random baseline, and the state-of-the-art methods AnyBURLExplainer (Betz et al., 2022), Data Poisoning (Zhang et al., 2019), and Kelpie (Rossi et al., 2022), using the Kinship (Kemp et al., 2006), WN18RR (Bordes et al., 2013), and FB15k-237 (Toutanova et al., 2015) KGs. Additionally, an empirical runtime evaluation of KGEPrisma and a qualitative evaluation of the three explanation types (instance-based, rule-based, analogy-based) are presented in this section. Furthermore, KGEPrisma's sensitivity in regard to its hyperparameters, its explanation robustness and limitations are discussed.

4.1 Evaluation Setting

The evaluation involves the three benchmark KGs FB15k-237, WN18RR, and Kinship designed for evaluating KGE models. FB15k-237 (Toutanova et al., 2015) is derived from FB15k to address the challenge of inverse relation test leakage. FB15k is built from the Freebase repository, which covers a wide range of domains such as music, films, locations, books, and people. FB15k's splits were problematic because many triples were inverses of each other, leading to leakage between training and testing datasets. FB15k-237 improves upon this by excluding inverse relations, thus containing 310,079 triples, 14,505 entities, and 237 relation types. The following evaluations used the standard train, test, and validation split.

WN18RR (Bordes et al., 2013) was developed as an improvement over the WN18 KG, a WordNet semantic network subset. WordNet describes semantic and lexical connections between terms, for instance, hyponyms, hypernyms, and antonyms. WN18RR was created to eliminate the issue of inverse relation leakage present in WN18. It includes 93,003 triples, maintaining the same set of 40,943 entities, but reduces the number of relation types to 11 from the original 18 in WN18. The following evaluations used the standard train, test, and validation split.

The Kinships (Kemp et al., 2006) KG maps the kinship relationships within the Alyawarra tribe from Australia. It comprises 10,686 triples, 104 entities, and 26 types of relations. The entities represent tribe members, and the relations are defined by specific kinship terms like Adiadya and Umbaidya, reflecting the social structure and rules of familial

ties within the tribe. Kinship has a high number of inverse relations, making it suboptimal for assessing the performance of KGE models. However, this enables the evaluation of whether the explainability method can capture the KGE model's dependency on inverse relations and its ability to pinpoint explanations in such regions of interest (e.g., $brotherOf(Tom, Hans) \rightarrow brotherOf(Hans, Tom)$). The following evaluations used the standard train, test, and validation split.

For the evaluation the three KGE models TransE, DistMult, ConvE are trained on the KG's. The three models were chosen as examples of KGE models with diverse interaction functions.

TransE (Bordes et al., 2013) is an energy-based model that produces embeddings by interpreting relationships as translations in a low-dimensional space (cf. Appendix A.1). If a relationship holds, the embedding of the tail entity should be proximal to the summation of the head entity embedding and the relation embedding.

DistMult (Yang et al., 2015) simplifies the RESCAL model (Nickel et al., 2011) by representing relationships with diagonal matrices instead of full matrices (cf. Appendix A.2). This reduction in computational complexity comes at the expense of expressive power, as DistMult cannot model anti-symmetric relations.

ConvE (Dettmers et al., 2018) utilizes a convolutional architecture, which includes a single convolution layer, a projection layer, and an inner product layer (cf. Appendix A.3). It is parameter-efficient and effective in modeling nodes with high in-degree, which is common in complex knowledge graphs.

These models were configured based on a comprehensive hyperparameter optimization study conducted by (Ali et al., 2021). Details about the embeddings they produce are provided in the Appendix A. The PyKEEN library (Ali et al., 2021) is used for implementing the KGE models.

The evaluation assesses KGEPrisma using the three different surrogate model configurations: MDI, K-Lasso, and HSIC-Lasso (cf. Sect. 3.4), all utilizing $k = 40$ for nearest neighbor search in step 1 (cf. Sect. 3.1). Additionally, the maximal clause length is set to $t = 2$ for FB15k-237, to $t = 3$ for WN18RR, and to $t = 1$ for Kinship in step 3 (cf. Sect. 3.3). This hyperparameter configuration performed best in an ablation study.

The evaluation of KGEPrisma is conducted against a simple retraining pipeline, two random baselines, AnyBURLExplainer (Betz et al., 2022), Data Poisoning (Zhang et al., 2019), and Kelpie (Rossi et al., 2022).

The retraining baseline retrains the KGE model without any changes to the training set. However, a different random seed may result in different outcomes.

The global random baseline selects a random path from the training KG as the explanation, which poses the risk of choosing irrelevant paths in large KGs. The local random baseline, however, selects a random path either starting or ending at the predicted head or tail node, utilizing triples close to the predicted triple, thereby impacting the prediction more significantly and creating a stronger baseline. Both baselines are adjusted to a path length of 2 for FB15k-237, of 3 for WN18RR, and of 1 for Kinship to align with KGEPrisma.

The state-of-the-art method by Betz et al. (2022) is used to compare the KGEPrisma against existing literature. The method by Betz et al. (2022) is called AnyBURLExplainer in the following. AnyBURLExplainer utilizes AnyBURL (Meilicke et al., 2019) to learn rules from a training knowledge graph, which are then used to attack and explain predicted

triples. The evaluation of AnyBURLExplainer is based on an implementation provided by its developers[3] limiting the rule length to 2 and the rule learning time to 100 s.

Data Poisoning (Zhang et al., 2019) is an adversarial attack method used for KGE models. The aim of this method is to manipulate the training set by adding or removing specific triples, which in turn alter the plausibility score assigned to a target triple by the KGE model. This manipulation is accomplished by shifting the embedding of the target triple based on the gradient of the KGE model's scoring function, with the goal of identifying adversarial triples. While data poisoning primarily serves as an adversarial attack method, the adversarial triples it generates can also be utilized as explanation triples. The implementation provided by Rossi et al. (2022) is used for this evaluation.[4]

Kelpie (Rossi et al., 2022) is an explainability framework designed for KGE models. It post-hoc identifies subsets of training facts that are either necessary or sufficient to explain specific predictions. Kelpie relies on heuristics such as holistic mimics and prefiltering to reduce the search space for explanations. A necessary explanation is the smallest set of facts such that removing these facts from the training set and retraining the model results in the prediction being false. A sufficient explanation is the smallest set of facts such that their addition to a set of entities enables the model to predict a specific tail of a query. The notion of faithfulness aligns with necessary explanations. Thus, necessary explanations produced by Kelpie are used for evaluation. The implementation of Kelpie provided by its developers is used for this evaluation.[4]

KGEPrisma, AnyBURLExplainer, and Kelpie provide multiple triples to explain one predicted triple. This allows for an evaluation protocol that captures the full expressiveness of the post-hoc XAI methods.

4.2 Faithfulness Evaluation Protocol

The evaluation protocol is designed to fully utilize an XAI method's expressivity, measuring its faithfulness by incorporating all triples that are part of the explanation. This approach ensures a comprehensive assessment of the method's performance.

Faithfulness (Hedström et al., 2023), in this context, measures the degree to which explanations mirror the predictive behavior of the KGE model, asserting that more crucial features have a stronger influence on model decisions. This relationship is traditionally verified in XAI through input perturbation, which means removing most relevant features and observing any decline in model performance (Hedström et al., 2023). However, input perturbation is not feasible for KGE models.

Instead, this paper proposes a **protocol** that includes training a KGE model on the complete training set D and selecting 30 optimal-performing triples P from the validation set, achieving a Hits@1 (Ali et al., 2021) and MRR (Ali et al., 2021) of 1 on P. An explanation E is generated for each triple in P, incorporating all grounding triples for the explanation rule or clause. These triples are then removed from D to form a new training set $D' = D \setminus E$, on which the model is retrained from scratch. The performance of P on this retrained model is assessed by Hits@1 and MRR, with values closer to 0 indicating a higher model deterioration and thus a more faithful explanation. As pointed out by Betz et al. (2022), it is crucial

[3] AnyBURLExplainer implementation: https://web.informatik.uni-mannheim.de/AnyBURL/.

[4] Data Poisoning and Kelpie implementation: https://github.com/AndRossi/Kelpie/tree/master.

that the filter set used to calculate Hits@1 and MRR remains unaltered by E to avoid artificially deflating model performance and to genuinely measure the impact of the explanation.

Figure 3 illustrates the evaluation protocol. We randomly select a set of 30 instances where the model performs optimally (Hits@1 = 1, MRR = 1). This number is chosen to be high enough to be representative yet low enough to remain computationally feasible given the retraining requirements. The random selection ensures that we capture a diverse and broad set of instances. Retraining is necessary because faithfulness measures the change in model predictions when the assumed "good reasons" supporting a prediction are removed. KGE models such as TransE or DistMult encode these good reasons during training. They are implicitly represented in all adjacent node and relation embeddings, which influence each other. Consequently, occluding these good reasons after training is not feasible, as the information is distributed throughout the learned embeddings. The only way to properly assess the impact of removing explanatory triples is to retrain the model without them, which is why retraining is an essential part of the evaluation protocol. This approach follows previous work on evaluating the faithfulness of XAI methods for knowledge graph completion models (Rossi et al., 2022; Betz et al., 2022).

4.3 Faithfulness Evaluation Results and Discussion

This section presents the evaluation results for the faithfulness evaluation protocol.

The results for **FB15k-237**, as detailed in Table 2, indicate that KGEPrisma consistently outperforms the other XAI methods across all KGE models.

For TransE, all methods demonstrate comparably poor performance, not meaningfully surpassing random baselines. The triples that TransE predicts on FB15k-237 are primarily self-loop relations, such as $/location/hud\_county\_place/place$, which connects the entity $Kansas\ City$ to itself, or the self-loop relation $/education/educational\_institution/campuses$, linking $Virginia\ Tech$ to itself. These self-loop relations are self-entailed and do not depend on patterns within the subgraph neighborhood. The model simply needs to remember them. This suggests that TransE's

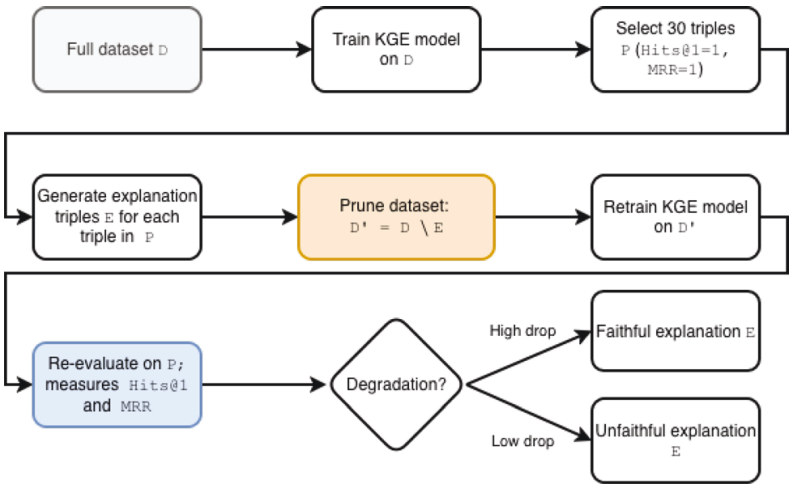

Fig. 3 Overview of the faithfulness evaluation protocol. Starting from the full training dataset D, a KGE model is trained and a set of 30 optimally predicted triples P (MRR = 1) is selected. For each triple in P, explanation triples E are generated using the XAI method under evaluation. The pruned dataset $D' = D \setminus E$ is then used to retrain the model. Finally, the model is re-evaluated on P: a high drop in Hits@1 and MRR indicates a faithful explanation, while a low drop suggests an unfaithful explanation

Table 2 Results for FB15k-237

	TransE		DistMult		ConvE	
	MRR	Hit@1	MRR	Hit@1	MRR	Hit@1
Retraining	0.97 ± 0.03	0.94 ± 0.06	0.84 ± 0.12	0.78 ± 0.16	0.84 ± 0.07	0.71 ± 0.00
Global random	0.96 ± 0.03	0.93 ± 0.05	0.82 ± 0.15	0.69 ± 0.27	0.76 ± 0.22	0.54 ± 0.43
Local random	0.96 ± 0.03	0.93 ± 0.04	0.70 ± 0.24	0.51 ± 0.36	0.69 ± 0.38	0.64 ± 0.43
AnyBURLExplainer	0.95 ± 0.02	0.92 ± 0.05	0.41 ± 0.35	0.25 ± 0.26	0.40 ± 0.25	0.16 ± 0.29
Data poisoning	0.96 ± 0.03	0.92 ± 0.05	0.40 ± 0.37	0.26 ± 0.29	0.41 ± 0.31	0.22 ± 0.36
Kelpie	**0.94** ± 0.03	**0.90** ± 0.06	0.37 ± 0.38	0.27 ± 0.33	0.49 ± 0.32	0.30 ± 0.36
KGEPrisma - K-Lasso	0.95 ± 0.04	0.92 ± 0.06	0.34 ± 0.08	**0.00** ± 0.00	0.41 ± 0.37	0.28 ± 0.41
KGEPrisma - HSIC-Lasso	0.95 ± 0.02	0.92 ± 0.04	**0.00** ± 0.00	**0.00** ± 0.00	**0.11** ± 0.09	**0.00** ± 0.00
KGEPrisma - MDI	0.95 ± 0.03	0.91 ± 0.05	**0.00** ± 0.00	**0.00** ± 0.00	0.32 ± 0.32	0.19 ± 0.37

The results are the mean and variance of the MRR and Hits@1 over ten runs; the lower the MRR and Hits@1, the better. The best results are bold

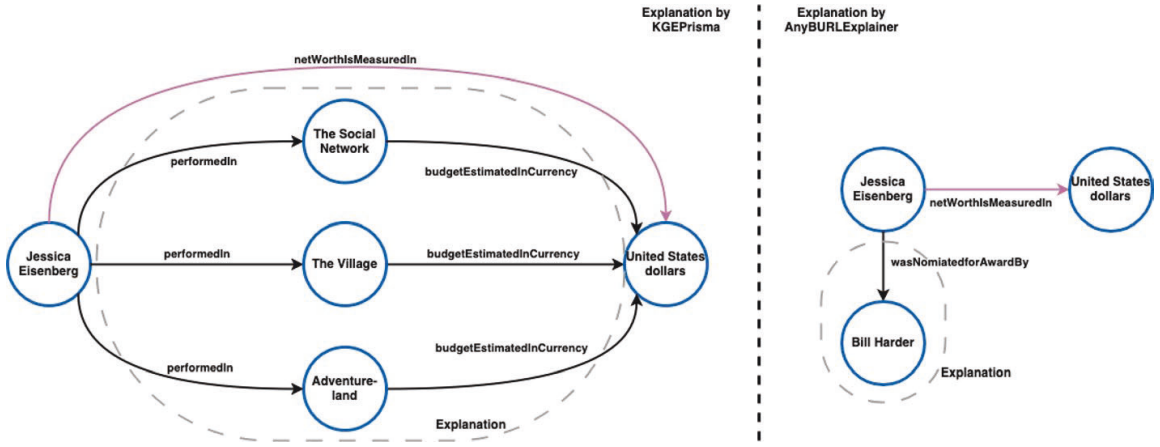

Fig. 4 DistMult in FB15k-237 predicts that Jessica Eisenberg's net worth is measured in US dollars. KGEPrisma explains this by pointing to the movies Jessica Eisberg performed in and their budget currency, US dollars, which is a sensible explanation. AnyBURLExplainer points to the fact that Jessica Eisberg was nominated alongside an award with Bill Hader as an explanation for Jessica Eisenberg's net worth being measured in US dollars. This explanation is not sensible

simplistic interaction function fails to effectively capture logical rules or patterns in FB15k-237. Instead, it relies on reconstructing relations from memory, which primarily works for self-loop relations and is hard to capture via triple-based explanations.

In the case of DistMult, KGEPrisma demonstrates the best performance compared to the other methods. Particularly noteworthy is its low variance. KGEPrisma with K-Lasso performs on par with other methods. However, when paired with HSIC-Lasso and MDI surrogate, KGEPrisma performs exceptionally well, deteriorating DistMult's MRR and Hits@1 to zero after retraining. This success can be attributed to KGEPrisma's ability to consistently identify better explanations.

For instance, DistMult predicts that $Jessica\ Eisenberg's$ net worth is measured in (i.e. relation: $/base/schemastaging/person\_extra/net\_worth./measurement\_unit/dated\_money\_value/currency$) $United\ States\ dollars$ (cf. Fig. 4), which is true. KGEPrisma explains this prediction by stating that $Jessica\ Eisenberg$ performed in the films (i.e. relation: */film/actor/film./film/performance/film*) $The\ Social\ Network$, *Adventureland*, and $The\ Village$. Since the budgets

for these films (i.e. relation: $/film/film/estimated\_budget./measurement\_unit/dated\_money\_value/currency$) are measured in $United\ States\ dollars$, it is quite plausible that $Jessica\ Eisenberg's$ net worth is also measured in $United\ States\ dollars$. In contrast, other XAI methods provide seemingly unrelated explanations. For example, AnyBURLExplainer explains the same predicted triple by stating that $Jessica\ Eisenberg$ was nominated (i.e. relation: $/award/award\_nominee/award\_nominations./award/award\_nomination/award\_nominee$) alongside $Bill\ Hader$ for an award, which appears irrelevant to the predicted triple. This disconnection in the predicted triple versus explanation is further reflected in their high variability in faithfulness scores.

In the case of ConvE on FB15k-237, the results are more mixed. Here, KGEPrisma with HSIC-Lasso outperforms the other methods by a significant margin. The other XAI methods, besides KGEPrisma with HSIC-Lasso, show similar performance, but they exhibit high variance, indicating inconsistency in their ability to provide reliable explanations. Looking at concrete examples reveals that all methods are able to identify high-impact explanation triples. However, KGEPrisma with HSCI-Lasso is the most consistent over the 10 runs in identifying such high-impact explanation triples. Thus, similar to the results for DistMult, KGEPrisma with HSIC-Lasso consistently generates more sensible explanations for ConvE on FB15k-237.

The overall strong performance of KGEPrisma on FB15k-237 across all models can be attributed to its capacity to reconstruct the decision surface of the KGE model locally. This allows it to produce explanation subgraphs that are highly aligned with the model's decision-making process. A similar effect can be observed on WN18RR.

The results for **WN18RR** in Table 3 also indicate state-of-the-art performance of KGEPrisma. This further emphasizes the importance of localizing explanations around the triples that the model identifies as similar (see Fig. 2, Step 1).

For the TransE model, KGEPrisma outperforms the other XAI methods across all surrogates, with K-Lasso being the best-performing surrogate. However, the variance indicates that the performance differences among the three surrogates are minimal.

Examining the explanations generated by KGEPrisma and the second-best performing method, AnyBURLExplainer, reveals that both can generally identify the same explanations. For instance, they consistently explain the predicted *hypernym* relationship between the entities *Platichthys* and $fish\_genus$ using its inverse, the $member\_meronym$

Table 3 Results for WN18RR

	TransE		DistMult		ConvE	
	MRR	Hit@1	MRR	Hit@1	MRR	Hit@1
Retraining	0.86 ± 0.04	0.78 ± 0.07	0.92 ± 0.08	0.88 ± 0.12	0.74 ± 0.28	0.62 ± 0.31
Global Random	0.75 ± 0.20	0.61 ± 0.27	0.91 ± 0.04	0.86 ± 0.06	0.65 ± 0.25	0.57 ± 0.23
Local Random	0.76 ± 0.22	0.65 ± 0.28	0.92 ± 0.03	0.87 ± 0.05	0.67 ± 0.28	0.58 ± 0.28
AnyBURLExplainer	0.26 ± 0.06	0.16 ± 0.06	**0.22** ± 0.05	**0.14** ± 0.04	0.17 ± 0.13	0.02 ± 0.07
Data Poisoning	0.42 ± 0.20	0.32 ± 0.23	0.39 ± 0.05	0.33 ± 0.06	0.07 ± 0.05	**0.00** ± 0.00
Kelpie	0.37 ± 0.11	0.28 ± 0.11	0.23 ± 0.05	0.16 ± 0.05	0.08 ± 0.05	**0.00** ± 0.00
KGEPrisma - K-Lasso	**0.05** ± 0.05	**0.02** ± 0.06	0.37 ± 0.07	0.30 ± 0.08	0.06 ± 0.11	**0.00** ± 0.00
KGEPrisma - HSIC-Lasso	0.10 ± 0.05	0.04 ± 0.04	0.38 ± 0.04	0.31 ± 0.04	**0.01** ± 0.01	**0.00** ± 0.00
KGEPrisma - MDI	0.15 ± 0.10	0.10 ± 0.11	0.26 ± 0.06	0.20 ± 0.05	**0.01** ± 0.01	**0.00** ± 0.00

The results are the mean and variance of the MRR and Hits@1 over ten runs; the lower the MRR and Hits@1, the better. The best results are bold

relationship. Nonetheless, there are cases where the inverse $member\_meronym$ relation does not exist to reconstruct the *hypernym* relationship due to the incompleteness of the knowledge graph. In such instances, AnyBURLExplainer fails to find an explanation, while KGEPrisma can construct more extensive and complex explanation chains, as can be observed with the *hypernym* relationship between the entity $family\_Treponemataceae$ and $bacteria\_family$. The correct explanation triples look as follows:

$$(order\_Spirochaetales,\ member\_meronym,\ family\_Treponemataceae),$$
$$(division\_Eubacteria,\ member\_meronym,\ order\_Spirochaetales),$$
$$(division\_Eubacteria,\ member\_meronym,\ bacteria\_family)$$

This explanation shows that the Treponemataceae family is a meronym of the Spirochaetales order, and the Spirochaetales order is a member of the Eubacteria division which in turn also has the bacteria family as a member. As a result, the bacteria family is also a hypernym of Treponemataceae. KGEPrisma successfully identified this complex relationship due to its localized approach to explanation, while AnyBURLExplainer was unable to do so.

For DistMult on WN18RR, AnyBURLExplainer performs the best. However, due to the variance in results, it is comparable to KGEPrisma using the MDI surrogate and Kelpie. When examining actual explanation examples, AnyBURLExplainer, KGEPrisma and Kelpie tend to produce similar explanations.

For ConvE on WN18RR, a similar trend can be observed. KGEPrisma shows the best performance, although it only marginally outperforms the other methods. This slight advantage is primarily due to KGEPrisma being more consistent in identifying the relevant explanation triples, as indicated by the variance and validated by the example explanations.

Once again, KGEPrisma outperforms AnyBURLExplainer, Kelpie, and Data Poisoning due to its localization around the predicted triple, consistently leading to accurate explanations. This observation also holds true when looking at the last benchmark knowledge graph, Kinship.

In the **Kinship** KG (cf. Table 4), KGEPrisma performs best for two of the three KGE models. For the TransE model, Kelpie, as well as Data Poisoning, outperforms KGEPrisma and AnyBURLExplainer. Looking at example predictions and explanations reveals why. The adversarial methods Kelpie and Data Poisoning robustly identify existing inverse relations, as what TransE relies on the most to reconstruct missing links. For example, the reconstruction of the relation *term*2 works consistenly with *term*1 as its inverse. The other XAI Methods have a higher chance of failing to identify such inverse relations, meanwhile coming up with different patterns which are less faithful to the model behaviour while still giving a sensible justification for why the link was predicted. This can also be observed in the still acceptable performance of KGEPrisma and AnyBURLExplainer in terms of faithfulness. For DistMult, all models perform comparatively well. KGEPrisma and AnyBURLExplainer share place one. KGEPrisma with the HSIC-Lasso surrogate, though, exhibits a slightly lower variance. The example predictions and explanations also show that they mostly come up with the same explanations. For example, for the prediction $(person23,\ term15,\ person29)$, both find the explanation $(person29,\ term5,\ person23)$. The faithfulness results for ConvE show that it is harder to explain its model behaviour. All models outperform the retraining and random baselines. However, after removing the explanations from the training graph, ConvE is still, even in the case of the most faithful explanations by KGEPrisma, able to

Table 4 Results for Kinship

	TransE		DistMult		ConvE	
	MRR	Hit@1	MRR	Hit@1	MRR	Hit@1
Retraining	0.89 ± 0.08	0.75 ± 0.09	0.64 ± 0.13	0.55 ± 0.07	0.94 ± 0.05	0.92 ± 0.04
Global Random	0.83 ± 0.03	0.77 ± 0.05	0.57 ± 0.12	0.49 ± 0.08	0.86 ± 0.07	0.78 ± 0.10
Local Random	0.80 ± 0.06	0.73 ± 0.09	0.53 ± 0.13	0.045 ± 0.03	0.83 ± 0.05	0.75 ± 0.08
AnyBURLExplainer	0.60 ± 0.07	0.33 ± 0.12	**0.04** ± 0.02	**0.00** ± 0.01	0.72 ± 0.06	0.57 ± 0.08
Data Poisoning	**0.05** ± 0.03	**0.01** ± 0.04	0.10 ± 0.04	0.04 ± 0.04	0.75 ± 0.04	0.61 ± 0.06
Kelpie	**0.05** ± 0.03	**0.01** ± 0.02	0.10 ± 0.06	0.03 ± 0.05	0.72 ± 0.05	0.56 ± 0.08
KGEPrisma - K-Lasso	0.52 ± 0.07	0.28 ± 0.09	0.05 ± 0.03	**0.00** ± 0.00	**0.65** ± 0.09	**0.50** ± 0.12
KGEPrisma - HSIC-Lasso	0.58 ± 0.07	0.33 ± 0.11	**0.04** ± 0.01	**0.00** ± 0.00	0.72 ± 0.07	0.57 ± 0.10
KGEPrisma - MDI	0.51 ± 0.06	0.23 ± 0.07	0.08 ± 0.04	0.03 ± 0.05	0.70 ± 0.08	0.55 ± 0.10

The results are the mean and variance of the MRR and Hits@1 over ten runs; the lower the MRR and Hits@1, the better. The best results are bold

reconstruct the predictions with a Hit@1 of 50%. Looking at the example predictions and explanations gives no visual clue as to why this is the case. The explanations of triples look comparable to the ones observed for the other KGE models.

Overall, KGEPrisma performs strongly for all models and KG's, demonstrating its ability to explain a wide variety of KGE models robustly.

4.4 Runtime Evaluation

KGEPrisma also performs well, runtime-wise, compared to the other XAI methods.

This can be observed, for example, in the benchmark knowledge graph Kinship. Figure 5 shows the mean runtime in seconds over ten runs of the XAI methods AnyBURLExplainer, Kelpie, Data Poisoning and KGEPrisma, explaining TransE, DistMult and ConvE in Kinship.

The setup and implementation used is the same as in the faithfulness evaluation. The runtime experiment is executed on an AWS EC2 g5.12xlarge.[5]

The results show that KGEPrisma calculates explanations in Kinship for all KGE models by a factor of approximately 10 to 20 times faster compared to AnyBURLExplainer and approximately 50 to 200 times faster compared to Kelpie. On the one hand, this performance gain is due to KGEPrisma's efficient localisation approach, approximating the KGE model's local decision surface by looking at the subgraphs sampled around triples the model learned to see as similar as described in Subsects. 3.1 to 3.4 (cf. Fig. 2 Step 1-3). Approximating the local decision surface results in a compact learning space for the surrogate model and, thus, a quick run-time. Kelpie, on the other hand, relies on expensive perturbation over triples to calculate an explanation. The perturbation is guided by a heuristic to focus on likely relevant

[5] Link to EC2 instance description: https://aws.amazon.com/de/ec2/instance-types/g5.

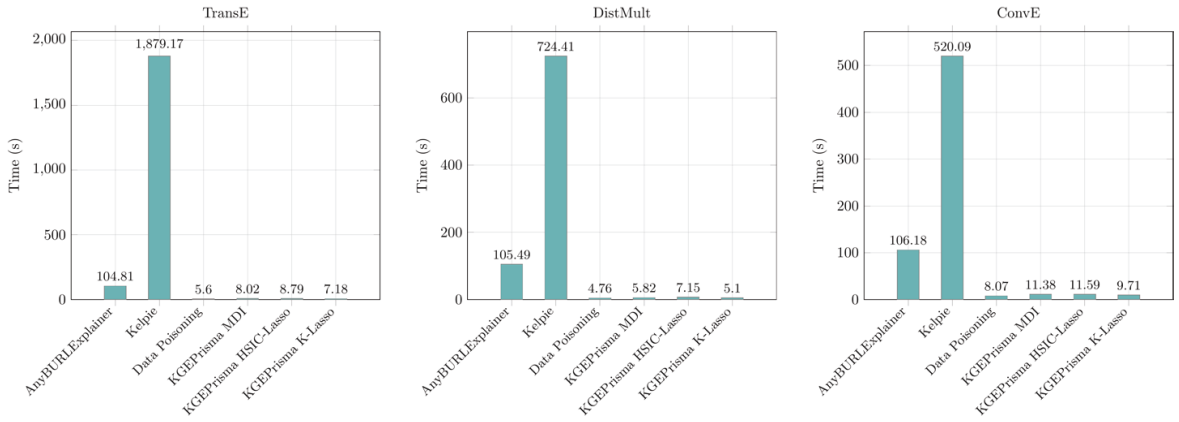

Fig. 5 Mean execution time (over 10 runs on the same hardware, 40 explanations per run) for AnyBURLExplainer, Kelpie, Data Poisoning, and KGEPrisma explaining TransE, DistMult, and ConvE in the Kinship KG

triples. However, this process is still runtime-wise expensive. AnyBURLExplainer has a constant run-time for all KGE models, as it does not depend on the underlying KGE model to calculate its explanations. Its execution time is set by the authors to a constant, meaning that it will return the best explanation found within the set timeframe (Betz et al., 2022).

Overall, the results demonstrate that KGEPrisma is a time-efficient method to post-hoc explain KGE models while delivering state-of-the-art faithfulness values.

4.5 Qualitative Discussion of the Three Explanation Modalities

The previous evaluation demonstrated that KGEPrisma performs at a state-of-the-art level in faithfulness while maintaining efficient runtime. Additionally, KGEPrisma is capable of generating three distinct types of explanations, rule-based, analogy-based, and instance-based, which present multiple perspectives on a single predicted triple to a user. The purpose of these different explanation types is to explain a single prediction from various perspectives to the user. This approach aims to ensure a better-informed user and, ultimately, to foster greater trust in the predicted link. The following qualitative evidence illustrates the effectiveness of these explanation modalities from the user's perspective.

Consider, for example, the prediction from the WN18RR KG that classifies the bacteria family as a hypernym of the Treponemataceae family. The instance-based explanation (cf. Fig. 6) for this prediction is provided in the form of a series of triples:

$$\begin{aligned}&(order\_Spirochaetales,\ member\_meronym,\ family\_Treponemataceae),\\&(division\_Eubacteria,\ member\_meronym,\ order\_Spirochaetales),\\&(division\_Eubacteria,\ member\_meronym,\ bacteria\_family).\end{aligned}$$

This explanation demonstrates that the Treponemataceae family is a subset (i.e., a meronym) of the Spirochaetales order, which in turn belongs to the Eubacteria division, a group that also includes the broader category referred to as the bacteria family. In this way, the instance-based explanation clarifies why the bacteria family can be seen as a hypernym of the Treponemataceae family.

While the instance-based modality provides focused context relevant to the specific prediction, it is limited by the narrow scope of the predicted triple itself. To capture a more gen-

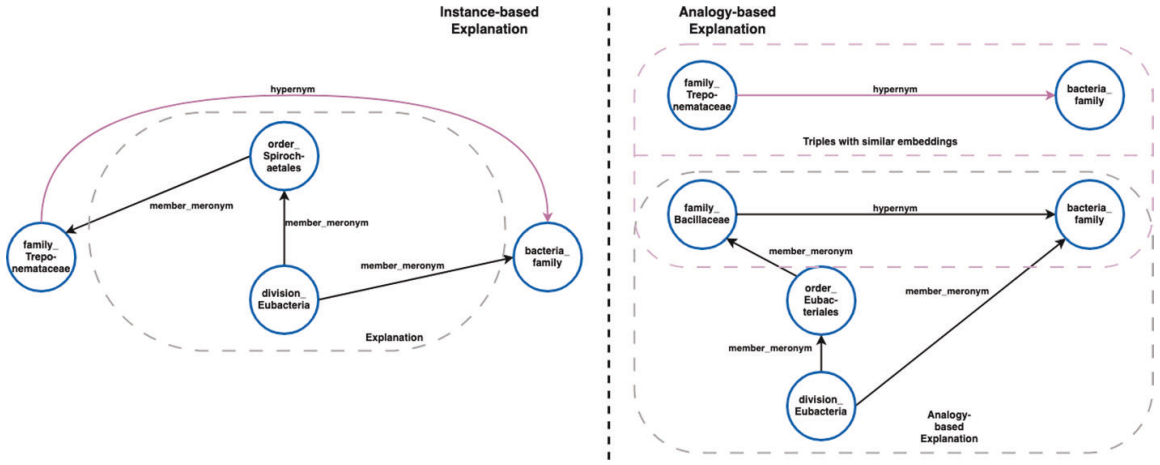

Fig. 6 TransE in WN18RR predicts that the bacteria family serves as a hypernym for the Treponemataceae family. An instance-based explanation provided by KGEPrisma explains this relationship by indicating that the Treponemataceae family is a subset (or meronym) of the Spirochaetales order. This order belongs to the Eubacteria division, which also includes the bacteria family. Thus, the instance-based explanation effectively demonstrates why the bacteria family can be considered a hypernym for the Treponemataceae family. Supporting the instance-based explanation, the analogy-based explanation provided by KGEPrisma points out to the user that the triple $(family\_Bacillaceae, hypernym, bacteria\_family)$ is similar to the predicted triple. In the context of this triple, we know that the Bacillaceae family is a subset of the Eubacteriales order, which in turn belongs to the Eubacteria division that includes the bacteria family. This analogous example reinsures the user that the reasoning pattern leading to the prediction aligns with other established facts within the KG

eral pattern within the domain, the user can look at the rule-based explanation. For example, the rule-based explanations for the same predicted triple states states:

$$\begin{aligned} hypernym(family\_Treponemataceae, bacteria\_family) \leftarrow \\ member\_meronym(X \in noun.animal, family\_Treponemataceae) \wedge \\ member\_meronym(Y \in noun.animal, X) \wedge \\ member\_meronym(Y, bacteria\_family). \end{aligned}$$

The rule states that the Treponemataceae family is hypernymic to the bacteria family, because it appears in a recurrent pattern that connects both entities, as evidenced by multiple instances across the knowledge graph. Such a rule indicates that the reasoning behind the prediction is not a one-off occurrence but rather reflects a recurring pattern in the domain, thereby reinforcing user confidence.

In addition, an analogy-based explanation (cf. Fig. 6) provides further support by comparing the prediction with another known instance. The knowledge graph embedding model learned in WN18RR that $family\_Bacillaceae$ is closely positioned to $family\_Treponemataceae$ in the latent space. The analogous explanation then outlines a similar set of relationships:

$$\begin{aligned} (order\_Eubacteriales,\ member\_meronym,\ family\_Bacillaceae), \\ (division\_Eubacteria,\ member\_meronym,\ order\_Eubacteriales), \\ (division\_Eubacteria,\ member\_meronym,\ bacteria\_family). \end{aligned}$$

This analogous case reaffirms that the reasoning pattern resulting in the prediction is consistent with other established facts within the knowledge graph.

Collectively, these explanation modalities provide distinct vantage points. They improve the user's understanding of the predictions and increase trust in the model's outputs.

4.6 Robustness

A property of explanation methods is robustness (Hedström et al., 2023). Similar inputs should yield similar explanations (Hedström et al., 2023). For KGEPrisma, robustness follows from the method's design, which is grounded in the smoothness principle (Bengio et al., 2013) of KGE models.

KGEPrisma generates explanations by first identifying the *k*-nearest neighbors in the latent embedding space (Step 1). Triples with similar embeddings will therefore share a substantial overlap in their nearest neighbor sets. This overlap propagates through the subsequent steps. Similar neighbor sets lead to similar positive and negative entity pairs (Step 2), which in turn yield similar clause frequency distributions (Step 3). Consequently, the surrogate models (Step 4) are trained on similar tabular representations. This results in similar feature importances and similar explanations (Step 5).

More formally, let $(e_1^{head}, r, e_1^{tail})$ and $(e_2^{head}, r, e_2^{tail})$ be two triples with similar embeddings, i.e., $\|v_1 - v_2\| < \epsilon$ for some small $\epsilon > 0$. Due to the continuity of the nearest neighbor retrieval, the sets of neighbors N_1 and N_2 will have significant overlap. As the clause mining and surrogate model fitting steps are deterministic functions of these neighbor sets, the resulting explanations will be similar.

This robustness is an advantage of KGEPrisma's embedding-based approach over methods that rely on global rule mining or gradient-based perturbations, which may produce inconsistent explanations for similar inputs. Empirically, we observed this behavior in our experiments. Triples of the same relation type with similar head or tail entities consistently received explanations with overlapping clauses and similar rule structures.

While achieving state-of-the-art results and being robust, KGEPrisma is sensitive to hyperparameters and has several limitations.

4.7 Sensitivity

KGEPrisma is sensitive to two hyperparameters. Which are the number of nearest neighbors *k* and the maximum clause length *t*.

The hyperparameter *k* determines the size of the local neighborhood in the embedding space from which positive and negative entity pairs are constructed. We selected $k = 40$ using the elbow method (Thorndike, 1953), which identifies the point at which adding more neighbors yields diminishing returns in terms of capturing the local structure. This approach ensures that *k* is large enough to provide a representative sample of similarly embedded triples, and small enough to maintain locality and computational efficiency.

The maximum clause length *t* controls the depth of the relational patterns captured by KGEPrisma. We set $t = 2$ for FB15k-237, $t = 3$ for WN18RR, and $t = 1$ for Kinship. These values were informed by the hop lengths commonly used in path-based link prediction methods such as MINERVA (Das et al., 2018) and DeepPath (Xiong et al., 2017), which have established appropriate path lengths for different knowledge graphs. For Kinship, the

prevalence of inverse relations means that single-hop patterns are often sufficient to capture the relevant regularities. WN18RR, with its hierarchical structure of hypernym and meronym relations, benefits from longer paths to capture multi-step semantic relationships. FB15k-237, covers diverse domains and performs well with moderate path lengths. KGEPrisma's performance remained stable across reasonable variations of these hyperparameters in our preliminary experiments.

4.8 Limitations

Although KGEPrisma is compatible with most KGE models, the method requires the KGE model to operate in Euclidean space, as the distance function used in Step 1 (cf. Sect. 3.1) is defined explicitly for such spaces. Consequently, it cannot post-hoc explain models like MuRe (Balažević et al., 2019) and QuatE (Zhang et al., 2019b), which utilize non-Euclidean spaces. Future work shall explore modular distance functions, including functions for non-Euclidian spaces.

Additionally, the choice of the hyperparameter k that determines the number of nearest neighbors in Step 1 (cf. Sect. 3.1) influences the quality of the explanation. If set too high, it can lead to explanations that are not localized around the instance to be explained and include noise, reducing their accuracy and relevance. Ideally, k should be set large enough to encompass all embeddings within a cluster. The elbow method can be employed to determine the ideal value (Thorndike, 1953).

The runtime of KGEPrisma is sensitive to the node degree of the KG. Higher node degrees result in an increased number of paths for expansion in Step 3 (cf. Sect. 3.3), which significantly extends the runtime for nodes with high connectivity.

Lastly, KGEPrisma assumes that KGE models learn embeddings solely from the symbolic structure of the knowledge graph. This limits its applicability to models incorporating literals, such as textual data, leading to less faithful explanations for such KGE models.

5 Related Work

Explainable Artificial Intelligence is about mapping the input of a black-box model to its output. That way, XAI methods compute an explanation of the model's behaviour. The explanation is supposed to justify the model's output (Adadi & Berrada, 2018; Lipton, 2018), helping to identify risks and flaws in the black-box model.

A common approach to achieve this are attribution methods. Attribution methods assign a value to input features resembling how relevant they are to the output. Among the notable local post-hoc XAI methods is LIME (Ribeiro et al., 2016). It utilizes local surrogate models to approximate the behavior of black-box models around specific instances, thereby providing local interpretability. Extending this concept, SHAP (Lundberg et al., 2017) employs Shapley values to calculate attributions across all possible coalitions of features. Integrated Gradients (Sundararajan et al., 2017) calculates the path integral of gradients along the straight line from a baseline to the input, highlighting the contribution of each feature to the difference in output. LRP (Montavon et al., 2019) backpropagates the output to the input layer, redistributing relevance scores across layers to identify relevant features. DEEPLift (Shrikumar et al., 2017) compares activation's to a reference activation, allocat-

ing relevance scores based on the difference, thus observing the shift caused by each input feature. These methods effectively assign the contributions of input features in standard settings such as tabular, image, or textual data. However, they are not trivial to apply to KGE models due to the complex and latent nature of the input triple, where simple attribution to input dimensions offers minimal insights into the predictive mechanisms.

Several XAI methods have been adapted for KGE models that employ graph neural networks (GNNs), yet their application remains constrained by the specific architectures and mechanisms of GNNs. GraphLIME (Huang et al., 2023) leverages local perturbation and the HSIC-Lasso as a surrogate model to approximate decision surfaces within GNNs. PGMExplainer (Vu & Thai, 2020) generates a probabilistic graphical model that captures the Markov blanket of the target prediction, though it is computationally expensive due to its reliance on input perturbation. GraphLRP (Schnake et al., 2022) adapts LRP for GNNs by propagating relevance through their aggregation functions. These methods, designed around the unique operations of GNN-based KGEs, are not universally applicable across the broader spectrum of KGE models, which often do not use GNN-based frameworks.

For that reason, inherently interpretable KGE models were introduced. DistMult (Yang et al., 2015), for example, introduces a unified framework for learning entity and relation representations using neural embeddings, emphasizing the extraction of logical rules from relation embeddings, which is primarily focused on capturing relational semantics through matrix operations. IterE (Zhang et al., 2019a) extends this by iteratively learning embeddings and rules to complement the weaknesses of each method, particularly enhancing the embeddings of sparse entities through rule incorporation. ExCut (Gad-Elrab et al., 2020) similarly combines embeddings with rule mining but shifts its focus towards generating interpretable entity clusters and iteratively refining them with rules derived from embedding patterns. However, these approaches are tailored to one specific KGE model and do not generalise to other models.

One stream of work that is applicable to KGE models comes from research on adversarial attacks. The work of Pezeshkpour et al. (2019) introduces gradient-based adversarial attacks, emphasizing the identification of influential training facts to test model sensitivity and robustness. Similarly, Data Poisoning (Zhang et al., 2019) targets the robustness of KGEs to adversarial attacks by proposing strategies for data poisoning that directly manipulate the knowledge graph. The approach by Bhardwaj et al. (2021) uses model-agnostic instance attribution methods from interpretable machine learning to select adversarial deletions, focusing on data poisoning to influence KGE predictions. This contrasts with AnyBURLExplainer (Betz et al., 2022). Which uses rule learning and abductive reasoning to perform adversarial attacks independent of the model's internal workings, thus focusing on explanation via adversarial attacks. AnyBURLExplainer outperformed other adversarial methods in its evaluation study. KGEPrisma deviates from these methods by focusing not on crafting adversarial inputs to disrupt the model but on decoding and interpreting the latent representations created by KGE models.

Other work focuses on explaining KGE models with surrogate models. The work by Ruschel et al. (2024); Gusmão et al. (2018) and Polleti and Cozman (2019) involves surrogate models that attempt to decode the embeddings through global and local perspectives respectively, with the former utilizing context-aware heuristics and the latter focusing on neighborhood features without capturing multi-hop dependencies. Meanwhile, KGEx (Baltatzis & Costabello, 2023) leverages multiple training subsets to generate

surrogate models that provide explanations based on training data impact. The approach by Islam et al. (2022) incorporates rule mining with embedding learning, independent of the KGE model. Yet other work such as OxKBC (Nandwani et al., 2020) and CPM (Stadelmaier et al., 2019) generate human-understandable explanations through heuristic templates and context paths, respectively, but struggle with scalability and faithfulness to underlying KGE models. KE-X (Zhao et al., 2023) leverages information entropy for subgraph analysis, improving interpretability but lacking clear heuristics for reconstructing model behavior. FeaBI (Ismaeil et al., 2023) constructs interpretable vectors for entity embeddings to provide model explanations, while (Chandrahas et al., 2020) focuses on semantic interpretability through entity co-occurrence statistics. Additionally, Kelpie (Rossi et al., 2022) and KGExplainer (Tengfei et al., 2025) introduce perturbation-based frameworks requiring retraining, which is resource-intensive. In contrast to these methods, KGEPrisma decodes the latent representations of KGE models. It identifies symbolic regularities in the subgraph neighborhood of the predicted link to generate rule-based, instance-based and analogy-based explanations. This approach remains faithful to the model's behavior and is computationally inexpensive, as it does not depend on perturbing training data or retraining the model.

6 Conclusion

This paper presented KGEPrisma, a novel post-hoc explainable AI method explicitly designed for KGE models. Despite their utility in knowledge graph completion, KGE models face criticism due to their black-box nature. KGEPrisma directly decodes these models' latent representations by identifying symbolic patterns within the subgraph neighborhoods of entities with similar embeddings. By translating these patterns into human-readable rules and facts, the method provides clear, interpretable explanations that bridge the gap between the abstract representations and predictive outputs of KGE models. This work contributes a post-hoc and local explainable AI approach that requires no retraining of the KGE model, is faithful to model predictions and can adapt to various explanation styles (rule-based, instance-based and analogy-based). Extensive evaluations demonstrated that this method outperforms state-of-the-art approaches, offering a distinct advantage by remaining faithful to the underlying predictive mechanisms of KGE models. Future research will apply KGEPrisma to knowledge graph domains, such as the biomedical field, where explainability is critical. Here, interpretable results can improve decision-making and foster trust in AI-based predictions. By providing transparent insights into the patterns and rules guiding KGE models, KGEPrisma uncovers hidden decision-making patterns in KGE models. Let us make KGE models understandable and trustworthy in high-risk use cases.

Appendix A: Triple Embedding Extraction

The construction of triple embeddings depends on the specific interaction function i of the Knowledge Graph Embedding model. Generally, triple embeddings represent the learned interaction between the head entity (e^{head}), relation (r), and tail entity (e^{tail}) before aggregation into a scalar score. This stage retains the richest representation of the entity dynam-

ics. Below, the triple embeddings v_{triple} for the three evaluated KGE models are described in detail.

TransE

The interaction function for TransE is:

$$i(e^{head}, r, e^{tail}) = -\|v^{head} + v^r - v^{tail}\|_2 \tag{A1}$$

where $v^{head}, v^r, v^{tail} \in \mathbb{R}^n$ are the learned embeddings of e^{head}, r, and e^{tail}, respectively (Bordes et al., 2013). Since $i(e^{head}, r, t)$ contains no additional parameters, it does not introduce information beyond the input embeddings. The triple embedding, denoted as v_{triple}, consolidates the learned embeddings of e^{head}, r, and e^{tail} before scoring:

$$v_{triple} = [v^{head}; v^r; v^{tail}], \tag{A2}$$

where [;] denotes vector concatenation.

DistMult

The interaction function for DistMult is:

$$i(e^{head}, r, e^{tail}) = \sum_{i=1}^{n} v_i^{head} v_i^r v_i^{tail}, \tag{A3}$$

where $v^{head}, v^r, v^{tail} \in \mathbb{R}^n$ are the learned embeddings of e^{head}, r, and e^{tail}, respectively (Yang et al., 2015). As with TransE, this function has no learned parameters, merely aggregating v^{head}, v^r and v^{tail} into a score. The triple embedding for DistMult is similarly defined as:

$$v_{triple} = [v^{head}; v^r; v^{tail}]. \tag{A4}$$

ConvE

ConvE differs from TransE and DistMult as its interaction function incorporates learned parameters, enriching the information within the triple embedding (Dettmers et al., 2018). For input embeddings $v^{head}, v^r, v^{tail} \in \mathbb{R}^d$, ConvE first combines v^{head} and v^r into a matrix $\mathrm{A} \in \mathbb{R}^{2\times d}$, where the rows represent v^{head} and v^r, respectively. A is reshaped into $\mathrm{B} \in \mathbb{R}^{m\times n}$, splitting its rows to represent v^{head} and v^r. A set of 2D convolutional filters $\Omega = \{\omega_i \mid \omega_i \in \mathbb{R}^{r\times c}\}$ is applied to B to capture interactions between v^{head} and v^r. The resulting feature maps are reshaped and concatenated into a feature vector $\mathrm{v} \in \mathbb{R}^{|\Omega|\cdot r_c}$, which is then mapped into the entity space via a linear transformation:

$$v^{head,r} = \mathrm{v}^\top \mathrm{W}, \tag{A5}$$

where $W \in \mathbb{R}^{|\Omega| \cdot r_c \times d}$. Finally, the interaction function aggregates the enriched representation with v^{tail} through:

$$i(e^{head}, r, e^{tail}) = v^{head,r} v^{tail}. \tag{A6}$$

The enriched representation $v^{head,r}$ combines and processes v^{head} and v^r through convolution, normalization, and dense layers, capturing more information than the individual embeddings (Ali et al., 2021). Thus, the triple embedding for ConvE concatenates $v^{head,r}$ and v^{tail}:

$$v_{triple} = [v^{head,r}; v^{tail}]. \tag{A7}$$

Acknowledgements We thank Sony AI for funding the research and Ute Schmid for enabling the collaboration.

Author Contributions - Christoph Wehner: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review and editing, Visualization, Project administration. - Chrysa Iliopoulou: Investigation, Writing – review and editing. - Ute Schmid: Formal analysis, Writing – review and editing. - Tarek R. Besold: Conceptualization, Resources, Writing – review and editing, Supervision, Funding acquisition.

Funding Open Access funding enabled and organized by Projekt DEAL. This research was fully funded by Sony AI.

Data Availability No datasets were generated or analysed during the current study.

Code Availability The code developed for this study is currently under review by Sony AI and will be made available upon publication of the article.

Declarations

Conflict of interest The authors declare no conflict of interest.

Ethical approval and consent to participate Not applicable. This study did not involve human participants, animals, or data requiring ethical approval or consent.

Consent for publication All authors have read and approved the final manuscript and consent to its publication. This manuscript does not contain any individual person's data in any form.

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access, 6*, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Ali, M., Berrendorf, M., Hoyt, C. T., Vermue, L., Galkin, M., Sharifzadeh, S., Fischer, A., Tresp, V., & Lehmann, J. (2021). Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2021.3124805

Bahr, L., Wehner, C., Wewerka, J., Bittencourt, J., Schmid, U., & Daub, R. (2025). Knowledge graph enhanced retrieval-augmented generation for failure mode and effects analysis. *Journal of Industrial Information Integration, 45*, Article 100807. https://doi.org/10.1016/j.jii.2025.100807

Balažević, I., Allen, C. & Hospedales, T. (2019). Multi-relational poincaré graph embeddings. In *Proceedings of the 33rd international conference on neural information processing systems*. Curran Associates Inc., Red Hook, NY, USA.

Baltatzis, V. & Costabello, L. (2023). *KGEx: explaining knowledge graph embeddings via subgraph sampling and knowledge distillation*.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Betz, P., Meilicke, C. & Stuckenschmidt, H. (2022). Adversarial explanations for knowledge graph embeddings. In Raedt, L.D. (ed.) *Proceedings of the thirty-first international joint conference on artificial intelligence, IJCAI-22*, pp. 2820–2826. International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria. https://doi.org/10.24963/ijcai.2022/391 . Main Track.

Bhardwaj, P., Kelleher, J., Costabello, L. & O'Sullivan, D. (2021). Adversarial attacks on knowledge graph embeddings via instance attribution methods. In Moens, M.-F., Huang, X., Specia, L., Yih, S. W.-t. (eds.) *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 8225–8239. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic. https://doi.org/10.18653/v1/2021.emnlp-main.648 .

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems.* (Vol. 26). Curran Associates Inc.

Chandrahas, Sengupta, T., Pragadeesh, C. & Talukdar, P. (2020). Inducing interpretability in knowledge graph embeddings. In Bhattacharyya, P., Sharma, D. M., & Sangal, R. (eds.) *Proceedings of the 17th international conference on natural language processing (ICON)*, pp. 70–75. NLP Association of India (NLPAI), Indian Institute of Technology Patna, India. https://aclanthology.org/2020.icon-main.9

Das, R., Dhuliawala, S., Zaheer, M., Vilnis, L., Durugkar, I., Krishnamurthy, A., Smola, A. & McCallum, A. (2018). Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In: *6th International conference on learning representations, ICLR 2018*, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, pp. 1–18. OpenReview.net. https://openreview.net/forum?id=Syg-YfWCW.

Dettmers, T., Minervini, P., Stenetorp, P. & Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *AAAI'18/IAAI'18/EAAI'18*. AAAI Press, New Orleans, Louisiana, USA.

Eirich, J., Jäckle, D., Sedlmair, M., Wehner, C., Schmid, U., Bernard, J., & Schreck, T. (2023). Manuknowvis: How to support different user groups in contextualizing and leveraging knowledge repositories. *IEEE Transactions on Visualization and Computer Graphics, 29*(8), 3441–3457. https://doi.org/10.1109/TVCG.2023.3279857

Gad-Elrab, M. H., Stepanova, D., Tran, T.-K., Adel, H. & Weikum, G. (2020). ExCut: Explainable embedding-based clustering over knowledge graphs. In Pan, J. Z., Tamma, V., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (eds.) *The semantic web – ISWC 2020*, pp. 218–237. Springer, Athens, Greece (2020). https://doi.org/10.1007/978-3-030-62419-4_13 .

Gusmão, A. C., Correia, A. H. C., De Bona, G. & Cozman, F. G. (2018). *Interpreting embedding models of knowledge bases: a pedagogical approach*. arXiv preprint arXiv:1806.09504.

Hanawa, K., Yokoi, S., Hara, S. & Inui, K. (2021). Evaluation of similarity-based explanations. In *International conference on learning representations*. https://openreview.net/forum?id=9uvhpyQwzM_.

Hedström, A., Weber, L., Krakowczyk, D., Bareeva, D., Motzkus, F., Samek, W., Lapuschkin, S., & Höhne, M. M. M. (2023). Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research, 24*(34), 1–11.

Hoerl, A. E., & Kennard, R. W. (2000). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 42*(1), 80–86.

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C.N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2021). Knowledge graphs. *ACM Computing Surveys*. https://doi.org/10.1145/3447772

Huang, Q., Yamada, M., Tian, Y., Singh, D., & Chang, Y. (2023). Graphlime: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering, 35*(7), 6968–6972. https://doi.org/10.1109/TKDE.2022.3187455

Islam, M. K., Aridhi, S., & Smail-Tabbone, M. (2022). Negative sampling and rule mining for explainable link prediction in knowledge graphs. *Knowledge-Based Systems, 250*, Article 109083. https://doi.org/10.1016/j.knosys.2022.109083

Ismaeil, Y., Stepanova, D., Tran, T.-K., & Blockeel, H. (2023). Feabi: A feature selection-based framework for interpreting kg embeddings. In T. R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng, & J. Li (Eds.), *The semantic web - ISWC 2023* (pp. 599–617). Springer.

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2022). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems, 33*(2), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T. & Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st national conference on artificial intelligence - Volume 1*. AAAI'06, pp. 381–388. AAAI Press, Boston, Massachusetts, USA.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue, 16*, 31–57. https://doi.org/10.1145/3236386.3241340

Lundberg, S. M. & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in neural information processing systems* 30, pp. 4765–4774. Curran Associates, Inc., Long Beach, California, United States. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

Meilicke, C., Chekol, M. W., Ruffinelli, D. & Stuckenschmidt, H. (2019). Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the 28th international joint conference on artificial intelligence*. IJCAI'19, pp. 3137–3143. AAAI Press, Macao, China.

Montavon, G., Binder, A., Lapuschkin, S., Samek, W. & Müller, K.-R. (2019). *Layer-wise relevance propagation: An overview*, pp. 193–209. Springer. https://doi.org/10.1007/978-3-030-28954-6_10 .

Nandwani, Y., Gupta, A., Agrawal, A., Chauhan, M. S., & Singla, P. (2020). Mausam: Oxkbc: Outcome explanation for factorization based knowledge base completion. In *Automated knowledge base construction*. https://openreview.net/forum?id=nqYhFwaUj.

Nembrini, S., König, I. R., & Wright, M. N. (2018). The revival of the Gini importance? *Bioinformatics, 34*(21), 3711–3718. https://doi.org/10.1093/bioinformatics/bty373

Nickel, M., Tresp, V. & Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on international conference on machine learning*. ICML'11, pp. 809–816. Omnipress, Madison, WI, USA.

Pezeshkpour, P., Tian, Y. & Singh, S. (2019). Investigating robustness and interpretability of link prediction via adversarial modifications. In Burstein, J., Doran, C., Solorio, T. (eds.) *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: human language technologies, Volume 1 (Long and Short Papers)*, pp. 3336–3347. Association for Computational Linguistics, Minneapolis, Minnesota. https://doi.org/10.18653/v1/N19-1337 .

Polleti, G. & Cozman, F. (2019). Faithfully explaining predictions of knowledge embeddings. In *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, pp. 892–903. SBC, Porto Alegre, RS, Brasil. https://doi.org/10.5753/eniac.2019.9343.

Ribeiro, M. T., Singh, S. & Guestrin, C. (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*, San Francisco, CA, USA, August 13-17, 2016, pp. 1135–1144.

Rossi, A., Barbosa, D., Firmani, D., Matinata, A., & Merialdo, P. (2021). Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data*. https://doi.org/10.1145/3424672

Rossi, A., Firmani, D., Merialdo, P. & Teofili, T. (2022). Explaining link prediction systems based on knowledge graph embeddings. In *Proceedings of the 2022 international conference on management of data*. SIGMOD '22, pp. 2062–2075. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3514221.3517887 .

Ruschel, A., Colombini Gusmão, A., & Gagliardi Cozman, F. (2024). Explaining answers generated by knowledge graph embeddings. *International Journal of Approximate Reasoning, 171*, Article 109183. https://doi.org/10.1016/j.ijar.2024.109183

Schnake, T., Eberle, O., Lederer, J., Nakajima, S., Schütt, K. T., Müller, K.-R., & Montavon, G. (2022). Higher-order explanations of graph neural networks via relevant walks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(11), 7581–7596. https://doi.org/10.1109/TPAMI.2021.3115452

Schramm, S., Wehner, C., & Schmid, U. (2023). Comprehensible artificial intelligence on knowledge graphs: A survey. *Journal of Web Semantics, 79*, Article 100806. https://doi.org/10.1016/j.websem.2023.100806

Schwalbe, G., & Finzel, B. (2023). A comprehensive taxonomy for explainable artificial intelligence: A systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*. https://doi.org/10.1007/s10618-022-00867-8

Shrikumar, A., Greenside, P. & Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proceedings of the 34th international conference on machine learning - Volume 70*. ICML'17, pp. 3145–3153. JMLR.org, Sydney, NSW, Australia.

Stadelmaier, J. & Padó, S. (2019). Modeling paths for explainable knowledge base completion. In: Linzen, T., Chrupała, G., Belinkov, Y., Hupkes, D. (eds.) *Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, pp. 147–157. Association for Computational Linguistics, Florence, Italy. https://doi.org/10.18653/v1/W19-4816 .

Sundararajan, M., Taly, A. & Yan, Q. (2017). Axiomatic attribution for deep networks. In *Proceedings of the 34th international conference on machine learning - Volume 70*. ICML'17, pp. 3319–3328. JMLR.org, Sydney, Australia.

Tengfei, M., Xiang, S., Wen, T., Mufei, L., Jiani, Z., Xiaoqin, P., Yijun, W., Bosheng, S. & Xiangxiang, Z. (2025). Towards synergistic path-based explanations for knowledge graph completion: Exploration and evaluation. In *Proceedings of international conference on learning representations*, pp. 1–20.

Thorndike, R. L. (1953). Who belongs in the family? *Psychometrika, 18*(4), 267–276. https://doi.org/10.1007/BF02289263

Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P. & Gamon, M. (2015). Representing text for joint embedding of text and knowledge bases. In Màrquez, L., Callison-Burch, C., Su, J. (eds.) *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1499–1509. Association for Computational Linguistics, Lisbon, Portugal. https://doi.org/10.18653/v1/D15-1174 .

Trouillon, T., Welbl, J., Riedel, S., Gaussier, E. & Bouchard, G. (2016). Complex embeddings for simple link prediction. In Balcan, M. F., Weinberger, K. Q. (eds.) *Proceedings of The 33rd international conference on machine learning. proceedings of machine learning research*, vol. 48, pp. 2071–2080. PMLR, New York, New York, USA. https://proceedings.mlr.press/v48/trouillon16.html

Vu, M. N. & Thai, M. T. (2020). Pgm-explainer: probabilistic graphical model explanations for graph neural networks. In *Proceedings of the 34th international conference on neural information processing systems*. NIPS '20. Curran Associates Inc., Red Hook, NY, USA.

Wehner, C., Kertel, M. & Wewerka, J. (2023). Interactive and intelligent root cause analysis in manufacturing with causal bayesian networks and knowledge graphs. In *2023 IEEE 97th vehicular technology conference (VTC2023-Spring)*, pp. 1–7. https://doi.org/10.1109/VTC2023-Spring57618.2023.10199563

Wehner, C., Powlesland, F., Altakrouri, B. & Schmid, U. (2022). Explainable online lane change predictions on a digital twin with a layer normalized lstm and layer-wise relevance propagation. In Fujita, H., Fournier-Viger, P., Ali, M., Wang, Y. (eds.) *Advances and trends in artificial intelligence: Theory and practices in artificial intelligence*, pp. 621–632. Springer, Cham.

Xiong, W., Hoang, T. & Wang, W.Y. (2017). DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 564–573. Association for Computational Linguistics, Copenhagen, Denmark. https://doi.org/10.18653/v1/D17-1060 .

Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., & Sugiyama, M. (2014). High-dimensional feature selection by feature-wise kernelized lasso. *Neural Computation, 26*(1), 185–207.

Yang, B., Yih, W., He, X., Gao, J. & Deng, L. (2015). Embedding entities and relations for learning and inference in knowledge bases. In Bengio, Y., LeCun, Y. (eds.) *3rd International conference on learning representations*, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings. http://arxiv.org/abs/1412.6575.

Zhang, H., Zheng, T., Gao, J., Miao, C., Su, L., Li, Y. & Ren, K. (2019). Data poisoning attack against knowledge graph embedding. In *Proceedings of the 28th international joint conference on artificial intelligence*. IJCAI'19, pp. 4853–4859. AAAI Press, Macao, China.

Zhang, S., Tay, Y., Yao, L. & Liu, Q. (2019). Quaternion knowledge graph embeddings. In *Proceedings of the 33rd international conference on neural information processing systems*. Curran Associates Inc., Red Hook, NY, USA.

Zhang, W., Paudel, B., Wang, L., Chen, J., Zhu, H., Zhang, W., Bernstein, A. & Chen, H. (2019). Iteratively learning embeddings and rules for knowledge graph reasoning. In: *WWW*, pp. 2366–2377. ACM, San Francisco, CA, USA.

Zhao, D., Wan, G., Zhan, Y., Wang, Z., Ding, L., Zheng, Z., & Du, B. (2023). Ke-x: Towards subgraph explanations of knowledge graph embedding based on knowledge information gain. *Knowledge-Based Systems, 278*, Article 110772. https://doi.org/10.1016/j.knosys.2023.110772

Authors and Affiliations

Christoph Wehner[1,2] · Chrysa Iliopoulou[1] · Ute Schmid[2] · Tarek R. Besold[1]

✉ Christoph Wehner
christoph.wehner@uni-bamberg.de

Chrysa Iliopoulou
chrysa.iliopoulou@sony.com

Ute Schmid
ute.schmid@uni-bamberg.de

Tarek R. Besold
tarek.besold@sony.com

[1] Hakken, Sony AI, Avinguda del Portal de l'Ángel 40, 08002 Barcelona, Catalonia, Spain

[2] Cognitive Systems Group, University of Bamberg, An der Weberei 5, 96047 Bamberg, Bavaria, Germany