When Confidence Turns Against You: Exploiting Link Prediction Scores for Data Poisoning

Adel Memariani[1], Arnab Sharma[1], Michael Röder[1], and Axel-Cyrille Ngonga Ngomo[1]

Data Science Group (DICE), Heinz Nixdorf Institute, Department of Computer Science, Paderborn University, Germany
{adel.memariani,arnab.sharma,michael.roeder,axel.ngonga}@upb.de

Abstract. Knowledge graph embedding models map symbolic entities and relations into continuous vector spaces and are widely used for downstream tasks such as link prediction. However, most existing embedding models assume that the training graph is *clean* and free from any malicious manipulation. This assumption leaves the models vulnerable because adversarial corruption in training data can degrade the model's overall performance. In this work, we use a score-based approach for identifying the most influential triples, and show that perturbing a small subset of them leads to a noticeable performance drop. Our proposed strategy requires only access to the model's confidence scores about the plausibility of triples. This substantially reduces the access requirement compared with existing white-box attacks that rely on internal model parameters such as gradients. Therefore, our attack setting reflects a more practical and realistic threat scenario. We evaluate the proposed attack across multiple KGE models and benchmark datasets. Our results show that a non-targeted, score-based poisoning method can effectively reduce the performance of state-of-the-art knowledge graph embedding models.

Keywords: Knowledge Graph Embeddings · Non-targeted Attacks · Data Poisoning · Link Prediction.

1 Introduction

Knowledge graphs (KGs) organize real-world facts through entities and relations. They provide a structured foundation for reasoning and prediction. Due to their ability to represent factual and relational knowledge in a machine-interpretable form, KGs have been widely used in applications such as information retrieval [10], question answering [16], and other knowledge-intensive tasks. Predicting missing facts in KGs is commonly addressed through knowledge graph embedding (KGE) models, which learn a scoring function over observed triples during training. At prediction time, this function assigns plausibility scores to unobserved triples, enabling link prediction and knowledge graph completion [27,41]. KGE approaches estimate triple plausibility through learned representations of entities and relations. At the data level, however, learning

high-quality entity and relation representations typically requires training data that is sufficiently large, diverse, and representative of the underlying knowledge graph. Such broad concept coverage is often achieved through community-curated KGs, which aggregate factual information at scale [1,17,41].

Although broader coverage can improve embedding quality, incorrect facts in community-curated knowledge graphs can still lead to a performance drop in KGE models [23,20]. Yet, KGE models are typically trained and validated with the assumption that the underlying KG is noise-free. However, this discrepancy creates an evaluation gap, as reported performance may not reveal whether KGE models learn and reproduce misinformation from the training graph [17,23]. This concern is made more pronounced by recent studies showing that editable knowledge graphs are vulnerable to adversarial manipulation. In particular, works on vandalism [18,28] and data poisoning strategies [21,5,41] are two well-studied examples. These studies, in general, help to expose systematic weaknesses in KGE models and provide a foundation for developing more robust architectures and training schemes [29]. Considering the existing works, from the perspective of the attackers' goal, strategies for data poisoning fall into two main categories [30]: (i) *Targeted attacks* that manipulate specific facts to influence certain inferences [26,6,5,40]. (ii) *Non-targeted attacks* that corrupt training triples to reduce the overall performance [43,20]. Our main goal is to expose vulnerability patterns in link prediction and thereby support the development of more robust KGE models. Targeted attacks are suitable when the aim is to alter predictions for specific facts, but their effectiveness depend strongly on the selected target triples and may not reveal broader weaknesses in the link prediction task. We therefore adopt a non-targeted setting that examines how poisoning affects overall link-prediction performance. Considering the attackers' access and knowledge, attack strategies can be further categorized into (i) *White-box* attacks, in which the attacker has full access to the model's learned parameters, scoring function, and gradients with respect to the entity and relation embeddings [26,6,5,40]. (ii) *Black-box* approaches where the attacker only has access to the training graph [43,20]. White-box attacks require privileged access to model internals, which is often unrealistic in practice [30,43]. In realistic deployments, KGE models may be accessible only through prediction APIs responses which may expose plausibility scores alongside ranked candidate triples. We therefore study an attack strategy that relies only on these returned scores. The proposed approach identifies score regions containing influential training triples and perturbs them to reduce overall link prediction performance. Our approach provides a broadly applicable way to analyze vulnerabilities across different KGE architectures and datasets.

We make the following contributions: (i) We formulate a non-targeted data-poisoning setting for link prediction. (ii) We develop a score-based method for identifying influential training triples. (iii) We conduct an extensive empirical evaluation on five widely used benchmark knowledge graphs and eight state-of-the-art KGE models. Overall, our analysis consists of 9,792 experiments, allowing us to systematically assess the effectiveness of the proposed attack strategy.

2 Related Work

Research on poisoning KGEs has evolved in recent years. In one of the earliest works, Zhang et al. [41] introduced the first dedicated poisoning method for KGE models and showed how strategic triple additions and deletions can increase or decrease the plausibility of any chosen fact after retraining. Most of the existing data poisoning attack approaches on KGE models are white-box strategies. For instance, [41,6,26,5,40] assume full access to the KGE model and design targeted white-box attacks by adding or deleting a small set of triples to raise or lower the model's score for specific facts. The approach in [26] studied targeted white-box poisoning by searching for a single triple whose insertion or removal flips the prediction for a given target triple after retraining. In [5], relation inference patterns are exploited to construct targeted white-box attacks that reduce the confidence of chosen target facts. This is done by reinforcing carefully selected alternative triples that compete with the targeted triple. So far, only a few works considered black-box poisoning.

The approach introduced in Banerjee et al. [3] implemented targeted black-box attacks with a reinforcement-learning agent that selects poison triples based on observed changes in the target triple's score. The method in [4] proposes targeted black-box attacks that use abductive reasoning over the knowledge graph to identify and perturb subgraphs without accessing model internals. More recently, Yang et al. [39] introduced targeted black-box poisoning, where a pre-trained language model generates adversarial triples given only the knowledge graph data, without access to the underlying KGE parameters.

Complementary to the above-mentioned attacks, Betz et al. [4] studied adversarial explanations in a black-box setting, generating perturbations that yield misleading local explanations for target triples without requiring gradients or parameters. Note that the aforementioned attack approaches are often studied in targeted settings, where the attacker seeks to manipulate specific predictions rather than degrade overall link prediction performance. In contrast to these approaches, Zhao et al. [43], and Kapoor et al. [20] focused on untargeted data poisoning attacks. Specifically, Zhao et al. [43] performed untargeted poisoning via learned logical rules. Therein, they learn Horn rules that capture global regularities in the KG and then perturb the training graph to reduce the support and confidence of these rules. As a result, its effectiveness is closely tied to how well the KG's relational patterns can be captured by a relatively compact set of such rules. Our goal is broader, and we study vulnerabilities of ranking-based link prediction, without assuming that the KG admits an accurate or stable symbolic summary in the form of Horn rules. Moreover, our approach scores triples in a single forward pass using model's scores (raw logits). In contrast, Zhao et al. [43] requires a multi-stage rule-based pipeline that learns and extracts rules and then grounds them over the KG. This adds substantial computational cost and can become a bottleneck on large graphs. Kapoor et al. [20] on the other hand, proposed a more efficient model-agnostic poisoning strategy, which does not directly exploit the behavior of the target embedding model. Specifically, their approach uses a random selection approach to select the triples to per-

turb. This essentially leads to less effectiveness. Finally, a closely related study investigates robust link prediction under non-targeted perturbations rather than proposing an attack method [23].

In a related study, Wang et al. [35] used plausibility scores for membership inference to identify whether candidate triples were part of the training data and thereby expose potential data-privacy risks. In contrast, our non-targeted poisoning method uses score quantiles to select training triples for perturbation, thereby reducing overall link-prediction performance. Beyond knowledge graphs, Duddu et al. [15] studied privacy leakage in graph embedding models. In their membership attack, an adversary uses the confidence scores returned by a GNN node classifier to determine whether a node belonged to the training graph. Their setting differs from ours in both the data representation and the attack objective. They consider graphs composed of nodes, edges, and node features, whereas we study relational knowledge graphs represented by triples. Moreover, their attack uses node-classification confidence scores to expose membership information. Our method uses KGE plausibility scores to select training triples for non-targeted poisoning.

3 Background

Let $\mathcal{E}$ and $\mathcal{R}$ denote the sets of entities and relations, respectively. A knowledge graph is represented as a set of triples $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{E},\ r \in \mathcal{R}\}$. A KGE model maps entities and relations into a vector space $\mathbb{V}$ through embedding matrices $\mathbf{E} \in \mathbb{V}^{|\mathcal{E}|\times d}$ and $\mathbf{R} \in \mathbb{V}^{|\mathcal{R}|\times d}$. It then assigns a plausibility score to each triple using a scoring function $\phi_\theta : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R}$ where θ denotes the model parameters. Higher values of $\phi_\theta(h, r, t)$ indicate that the triple is considered more plausible, whereas lower values indicate that it is less plausible. In the non-targeted poisoning setting, the attacker does not aim to corrupt a specific triple. Instead, given a perturbation budget b, the goal is to select a small subset of training triples $\mathcal{B} \subseteq \mathcal{G}_{\text{train}}$, $|\mathcal{B}| = b$, whose perturbation leads to a *noticeable* degradation in overall link-prediction performance. Since candidates are selected from a large training graph, exhaustively searching over all possible subsets $\mathcal{B}$ is computationally infeasible. Therefore, practical poisoning strategies require selection criteria that can identify influential triples without evaluating all possible perturbation combinations.

Let $m(\theta; \mathcal{G}_{\text{test}})$ denote a link-prediction metric MRR, evaluated on the test graph, where larger values indicate better performance. Let θ_{clean} be the parameters learned from the original training graph $\mathcal{G}_{\text{train}}$, and let $\theta_{\mathcal{B}}$ be the parameters learned after perturbing the selected subset $\mathcal{B}$. We define the performance degradation induced by $\mathcal{B}$ as $\Delta_m(\mathcal{B}) = m(\theta_{\text{clean}}; \mathcal{G}_{\text{test}}) - m(\theta_{\mathcal{B}}; \mathcal{G}_{\text{test}})$. A poisoning attack is considered noticeable if the induced degradation exceeds a predefined threshold τ, i.e., $\Delta_m(\mathcal{B}) \geq \tau$. Equivalently, we can define noticeable degradation in relative terms defined as,

$$\frac{m(\theta_{\text{clean}}; \mathcal{G}_{\text{test}}) - m(\theta_{\mathcal{B}}; \mathcal{G}_{\text{test}})}{m(\theta_{\text{clean}}; \mathcal{G}_{\text{test}})} \geq \rho \tag{1}$$

where $\rho > 0$ denotes the minimum relative performance drop required for the attack to be considered effective.

3.1 Threat Model

We consider an adversary with score-query access to the victim link prediction system. Specifically, given a candidate triple, the attacker can obtain the corresponding real-valued output score (e.g., plausibility or confidence). The attacker has no access to the victim model's parameters, gradients, training objective, intermediate representations, or training losses, and cannot otherwise inspect or modify the model beyond issuing such queries. We assume the attacker operates under a limited perturbation budget to remain hard-to-detect by fact validation or noise detection mechanisms. Consequently, only a small fraction of the training graph can be modified, and smaller ratios are preferable in practice. We adapt the same poisoning setting as in [23,43,19]. The attacker cannot introduce new entities or relations. Under this setting, the attacker can perturb $k\%$ of existing triples by substituting the head, tail, or relation with another entity or relation drawn randomly from the same knowledge graph. More broadly, our approach considers settings in which knowledge graphs are continuously maintained and updated, and periodically reused to retrain link-prediction models. In such settings, an attacker may obtain plausibility scores through a prediction interface that returns scores for submitted queries. The attacker can then introduce a small number of selected perturbations into a later update. This threat is particularly relevant to open knowledge graphs, which allow external contributors to edit or add content. Once the KGE model is retrained, these perturbations affect subsequent predictions. Our surrogate-based attack serves as a proxy for this score-querying scenario. In our implementation, all training triples are scored in a single forward pass through the trained surrogate model.

4 Methodology

We aim to identify the most influential training triples such that modifying a small subset of them yields the largest drop in link prediction performance. Primarily, optimizing attacks against link prediction has a very high computational overhead, since evaluation requires repeated ranking over large candidate sets. To make influence estimation tractable, in this paper, we rely on the model's plausibility score for each triple. Concretely, we use the raw model's logits as the signal to prioritize which training triples are most influential for link prediction, i.e., those whose perturbation yields the largest drop in performance. Although our experiments use raw plausibility scores, the proposed attack scenario does not depend on their numerical scale. Triple selection relies only on the global ordering of the scores. A fixed monotonic transformation, such as the sigmoid function used to map model outputs to bounded values, preserves this ordering. The proposed scenario thus requires only a score signal that is comparable across triples and does not necessarily require access to raw logits. Figure 1 illustrates

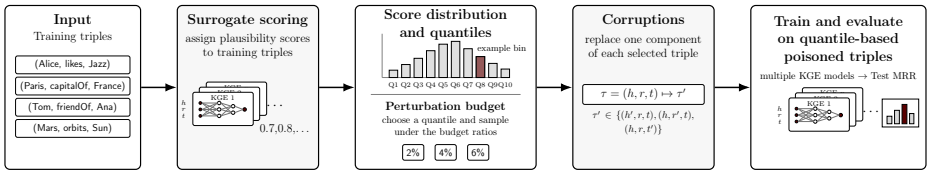

Fig. 1. Workflow of the score-based poisoning approach. A surrogate model assigns plausibility scores to training triples, which are partitioned into score quantiles. A target quantile is selected, and a fixed perturbation budget is used to modify the corresponding triples. The perturbed training graph is then used to train and evaluate multiple KGE models.

the overall workflow of our score-based poisoning approach. Starting from the original training triples, a surrogate KGE model assigns a plausibility score to each triple. The triples are then partitioned into score quantiles, from which a subset is selected according to a fixed perturbation budget. Each selected triple is corrupted by replacing one of its components, and the resulting poisoned graph is used to train and evaluate the victim KGE models in terms of test MRR. Below we describe the steps in detail.

4.1 Plausibility Scoring in KGE Models

Knowledge graph embedding models differ in their scoring functions but share a common learning principle. Given a triple $\tau = (h, r, t)$, a KGE model defines a scoring function $\phi_\theta : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R}$, where θ denotes the model parameters, including entity and relation embeddings and any model-specific transformations. During training, the model learns to assign higher scores to plausible triples than to implausible ones. Therefore, for a positive triple τ^+ and a corresponding corrupted triple τ^-, the training objective typically encourages $\phi_\theta(\tau^+) > \phi_\theta(\tau^-)$.

4.2 Surrogate-Based Triple Scoring

In a non-targeted attack setting, performance degrades most when perturbations hit the triples that the model strongly relies on for its relational inference. Our goal is therefore to identify these influential triples using a model-derived signal. Motivated by this principle, we train a surrogate model for 100 epochs, which was sufficient for the training objective to converge. We then perform a single forward pass over all triples in the knowledge graph to obtain the model's plausibility scores. Let $\hat{s}_\phi(h, r, t)$ denote the plausibility score returned to the attacker for a triple ((h,r,t)). In the score-query setting, the attacker obtains these scores by submitting triples to the victim model's prediction interface, without requiring access to the internal parameters that produces the final output. Because our experiments do not query a deployed victim model, we use a surrogate KGE model solely to simulate the plausibility scores that an attacker could obtain through a prediction interface. Each editable training triple $\tau \in \mathcal{G}$train is assigned a numerical plausibility score s_τ. The resulting scores are then used to order and group the triples, from which a subset is selected for perturbation. In

this regard, we interpret $\mathcal{S}$ as the empirical distribution of plausibility scores assigned by the model to the training triples. This distribution allows us to study how the effect of poisoning varies across different score ranges. In particular, low-scoring triples correspond to facts that the model does not strongly support, and high-scoring triples correspond to facts that the model strongly ranks as plausible. Intermediate-scoring triples correspond to facts that the model does not clearly distinguish as plausible or implausible. These score ranges reflect the relative position of triples in the model's score ordering. Accordingly, the training triples are grouped according to the quantile intervals of their plausibility scores, and the triples to be perturbed are selected from a chosen quantile group.

4.3 Quantile-Based Triple Selection

Let $T = \mathcal{G}_{\text{train}}$ be the set of triples in the knowledge graph. Each triple $t \in T$ is assigned a plausibility score s_t. The triples are then sorted in ascending order of their scores:

$$T^{\uparrow} = \big(t_{(1)}, t_{(2)}, \ldots, t_{(|T|)}\big), \qquad s_{t_{(1)}} \leq s_{t_{(2)}} \leq \cdots \leq s_{t_{(|T|)}}. \tag{2}$$

The position of a triple in this ordered list defines its empirical score quantile. For $a, u \in [0, 1]$ with $a < u$, we define the score-slice operator as:

$$\xi(T^{\uparrow}, a, u) = \big\{ t_{(i)} \;\big|\; \lfloor a|T| \rfloor < i \leq \lfloor u|T| \rfloor \big\} \tag{3}$$

This operator essentially returns all triples whose rank positions lie between the lower quantile boundary a and the upper quantile boundary u. Now we divide the ordered training triples into ten score-induced quantile bins. For $q \in \{1, \ldots, 10\}$, the q-th quantile bin is defined as $Q_q = \xi\left(T^{\uparrow}, \frac{q-1}{10}, \frac{q}{10}\right)$. Here, Q_1 contains the lowest-scoring 10% of triples, whereas Q_{10} contains the highest-scoring 10% of triples. Given a perturbation budget b, we sample the attack set from the chosen quantile bin as $C_q \sim \text{UniformSubset}(Q_q, b)$, where $|C_q| = b$. This essentially allows us to compare how perturbing triples from different regions of the score distribution affects downstream link-prediction performance. Now for each selected triple $\tau = (h, r, t) \in C_q$, the perturbed triple τ' is then defined as

$$\tau' = \begin{cases} (h', r, t), & z = \text{head}, \quad h' \sim \text{uniform}(\mathcal{E} \setminus \{h\}), \\ (h, r', t), & z = \text{relation}, \quad r' \sim \text{uniform}(\mathcal{R} \setminus \{r\}), \\ (h, r, t'), & z = \text{tail}, \quad t' \sim \text{uniform}(\mathcal{E} \setminus \{t\}). \end{cases} \tag{4}$$

For each selected triple, one corruption position is sampled uniformly from the head, relation, and tail positions and then perturbed. Let $P_q = \{\tau' \mid \tau \in C_q\}$ denote the set of corrupted triples generated from the selected attack set. The poisoned training graph is obtained by replacing the selected clean triples with the corrupted triples as $\widetilde{\mathcal{G}}^{(q)}_{\text{train}} = (\mathcal{G}_{\text{train}} \setminus C_q) \cup P_q$. This perturbation preserves the original entity and relation vocabularies and keeps the number of training triples unchanged. However, it alters the training data by replacing valid triples

with syntactically valid but semantically mismatched triples. By evaluating the model trained on $\widetilde{\mathcal{G}}^{(q)}_{\text{train}}$ for each $q \in \{1, \ldots, 10\}$, we can analyze which regions of the surrogate score distribution are most vulnerable to poisoning. In our proposed approach, although sampling within a score quantile is uniform, the overall procedure is not random because the quantile partitioning imposes a principled selection over the training triples. Sampling is used only to respect a limited perturbation budget. We adopted ten quantiles to show how effectiveness varies across the score distribution. In practice, the approach becomes more effective when focusing on the most influential region and using narrower ranges, where uniform sampling becomes less critical. We observed this effect empirically, but to keep the computational cost and runtime manageable, we used ten quantiles. Primarily, we evaluated our proposed attack strategy on standard benchmark knowledge graphs to ensure reproducibility, controlled experimentation, and fair comparison across embedding approaches. This means that we can (i) Fix the underlying graph. (ii) Apply a precisely specified poisoning budget. (iii) Retrain and evaluate across multiple settings. This enables building dataset- and method-specific profiles, and it yields actionable insights for assessing the robustness of link prediction tools under potential data-poisoning attacks.

5 Evaluation

We evaluate how poisoning effectiveness varies across the plausibility score distribution using eight state-of-the-art KGE models, five benchmark knowledge graphs that vary in size and density, and perturbation budgets corresponding to 2%, 4%, and 6% of the training triples.

5.1 Baseline:

We adopt the non-targeted perturbation strategy of Kapoor et al. [20] as our baseline. Since their approach may generate corrupted triples that already exist in the knowledge graph, we discard such perturbations and resample until the budget is filled with distinct new triples. This yields a controlled comparison, i.e., the baseline and our approach use the same corruption operator. We do not compare against the rule-based attack of Zhao et al. [43]. Their method requires an additional rule-learning stage, whose cost and effectiveness depend on the presence of usable logical patterns in the target graph. Our setting does not depend on this assumption. We do not rely on auxiliary signals, rule sets, ontologies, or prior assumptions about the knowledge graph's logical structure. Instead, our approach uses only the scores returned by the link-prediction models. Moreover, their implementation was not publicly available and a direct reproducible comparison was therefore not possible.

5.2 Datasets:

We conduct our experiments on five widely adopted knowledge graphs: (i) KINSHIP [13] represents family ties in a small anthropological domain. (ii) UMLS [22]

is a biomedical dataset that aggregates and standardizes medical terminologies. (iii) FB15K-237 [32] is derived from Freebase [7]. It encodes real-world entities and relations. (iv) NELL-995-h100 [37] is a curated subset of the Never-Ending Language Learning project. (v) WN18RR [14] is a refined benchmark derived from WordNet [24]. Table 1 provides an overview of the datasets. It shows the number of entities $|E|$ and the number of relations $|R|$ in each dataset. It also shows the number of triples in train, validation, and test datasets. We adopted *Entity Density (ED)* and *Relation Density (RD)* measures from [23] to quantify graph sparsity. Entity Density is defined as ED $= \frac{2|T|}{|\mathcal{E}|}$, and $|\mathcal{E}|$ the number of entities. Smaller ED indicates a sparser graph. Relation Density is defined as RD $= \frac{|T|}{|\mathcal{R}|}$, where $|\mathcal{R}|$ is the number of relations, and it reflects the average number of triples per relation. Based on these measures, WN18RR has the smallest number of triples per entity on average, and it is the most sparse dataset among others. NELL-995-h100 also has a low ED and is sparse. On the contrary, KINSHIP has the largest ED and is the most dense dataset, followed by UMLS and then FB15k-237 in terms of decreasing density.

Dataset	$\|E\|$	$\|R\|$	$\|Train\|$	$\|Val\|$	$\|Test\|$	ED	RD
KINSHIP	104	25	8,544	1,068	1,074	102.75	427.44
UMLS	135	46	5,216	652	661	48.36	141.93
FB15K-237	14,541	237	272,115	17,535	20,466	21.33	1308.51
NELL-995-h100	22,411	43	50,314	3,763	3,746	**2.58**	1344.72
WN18RR	40,943	22	86,835	3,034	3,134	**2.27**	8454.82

Table 1. Summary statistics of the datasets

5.3 Knowledge Graph Embedding Approaches

For our experiments, we study eight broadly used, state-of-the-art knowledge graph embedding models that represent current best-performing approaches for link prediction: (i) TransH [36] extends TransE [8] by performing the translation in a relation-specific hyperplane rather than in the shared embedding space. This design was introduced to reduce the limitations of TransE, specifically for 1-to-many, many-to-1, and reflexive relations. (ii) DistMult [38] is a parameter-efficient bilinear extension of the RESCAL [25] tensor factorization method. RESCAL uses per-relation parameters that can limit scalability and increase the risk of overfitting [11,33,2]. DistMult handles this by restricting each relation matrix to be diagonal. It performs well on symmetric relations, but it cannot model antisymmetric patterns. (iii) ComplEx [34] extends DistMult by using complex-valued embeddings for entities and relations. This allows ComplEx to capture antisymmetric relation patterns. (iv) QMult [42] extends DistMult by using quaternion-valued embeddings in $\mathbb{H}$. Its benefits tend to be more visible on larger knowledge graphs [11]. (v) DualE [9] is a hypercomplex KGE model. It represents both entities and relations in dual-quaternion space. This allows

each relation to encode a combined rotation and translation. (vi) Keci [12] is based on the assumption that the most appropriate embedding algebra varies with the characteristics of the target knowledge graph [23,12]. As a result, it provides a unifying framework that subsumes earlier models such as DistMult [38] and ComplEx [34]. (vii) RotatE [31] represents entities in a complex vector space and parameterizes relations as rotations. This formulation captures key relational patterns such as composition, inversion, and symmetry or antisymmetry [11]. It also scales efficiently to large graphs, with time and memory requirements that grow linearly with the dataset size [11]. (viii) MuRE [2] models relation-dependent hierarchies by learning separate transformations for head and tail embeddings. This is motivated by the fact that a single knowledge graph can induce multiple, potentially overlapping hierarchical structures over the same entities, depending on the relation type.

5.4 Results

To account for variability due to random initialization, we repeated every setting with six different random seeds. Altogether, we performed 9,792 experiments. This setup enables a direct comparison of perturbations made from different parts of the score distribution. For example, we can perturb triples selected only from low-score quantiles or only from high-score quantiles and quantify the resulting drop in overall link prediction performance. Detailed results for all dataset-model-budget combinations are provided in the supplementary material. For brevity, we focus here only the generic trend of the results. We observe that perturbations on denser datasets are generally more harmful when they target triples in the mid-to-high score ranges, corresponding to the upper score quantiles. However, for the sparse datasets, NELL-995-h100 and WN18RR, the same trend does not consistently hold. The impact of perturbation varies more across models and budgets, and larger performance drops often occur when lower score quantiles are perturbed. This indicates that the role of score regions depends on graph density, specifically, while mid-to-high scored triples are often the most damaging targets in denser graphs, low-scored triples can be more influential in sparse graphs. Possible explanations are discussed in Section 6.

5.5 Statistical Significance

In addition to the quantitative comparisons, we perform statistical tests to assess the significance of the results. Specifically, we apply a paired t-test to compare performance drops under the baseline against drops caused with our score-based perturbation strategy, using matched runs across the same seeds. We focus on cases where the score-based strategy yields a lower test MRR than the baseline and where the decrease is statistically significant. In our paired t-test analysis, we consider a difference statistically significant when the resulting p-value is below 0.05. Figure 2 highlights a clear pattern. The dark red cells highlight statistically significant performance drops. In the denser datasets (KINSHIP, UMLS, and FB15k-237), statistically significant degradations concentrate in the upper score

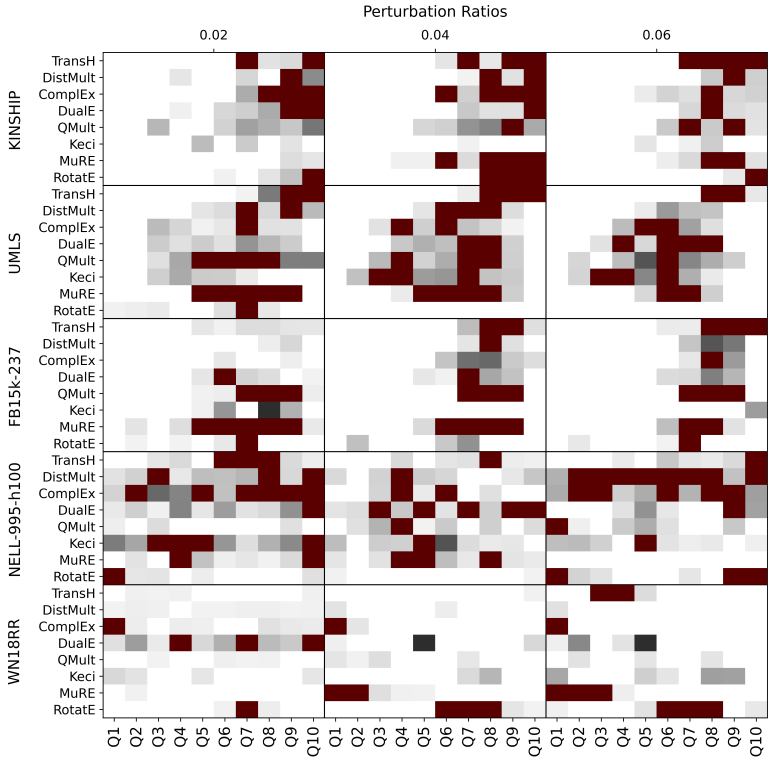

Fig. 2. Grey cells indicate cases where score-based poisoning yields a lower MRR than random corruption, with darker grey showing a larger decrease. **Dark red** cells highlight statistically significant performance drops.

quantiles. They occur most often when perturbations target mid-high to high-scored triples, while the highest-score quantile is comparatively less frequently significant. Results that are lower than baseline but not statistically significant are shown in gray, and the darker the gray, the larger the drop relative to the baseline. These parts follow the same trend, with larger decreases concentrating in the upper score quantiles for the denser datasets.

An additional implication of our statistical analysis is that score quantiles can substantially narrow the attacker's candidate space. For KINSHIP, UMLS, and FB15k-237, statistically significant performance drops concentrate in the upper score quantiles, while the first half of the quantiles rarely yields competitive drops under the same perturbation budget. In these datasets, an attacker can therefore exclude more than half of the training triples from consideration and focus on mid-high to high scored triples, which reduces the candidate pool and simplifies the selection problem. For NELL-995-h100 and WN18RR, the pattern shifts toward lower score quantiles, implying that search-space reduction is still possible but should prioritize low to mid-low scored triples instead.

5.6 Relative Performance Drop

Finally, for each dataset–model combination, we quantify the impact of poisoning as the relative reduction in test MRR caused by our score-based perturbation strategy and by the baseline strategy. For each dataset, KGE model, and perturbation budget b, the clean reference is the model trained on the original, unperturbed training graph. All experiments are repeated using the same set of random seeds. Let $\mathcal{Q} = 1, \ldots, 10$ denote the set of score quantiles, and let $\mathcal{B}b, q \subseteq \mathcal{G}\text{train}$ denote the set of b training triples selected from quantile q for perturbation. Let $\widetilde{\mathcal{B}}_{b,q}$ denote their corrupted versions. The corresponding perturbed training graph is defined as:

$$\mathcal{G}_{\text{train}}^{b,q} = (\mathcal{G}_{\text{train}} \setminus \mathcal{B}_{b,q}) \cup \widetilde{\mathcal{B}}_{b,q} \tag{5}$$

Thus, $\mathcal{G}\text{train}^{b,q}$ contains all unmodified training triples together with the b perturbed triples. For each random seed s, we first train the model on the clean training graph and obtain the test MRR m_s^{clean}. We then train the same model configuration on $\mathcal{G}\text{train}^{b,q}$ and obtain the test MRR $m_s^{b,q}$. The validation and test graphs remain unchanged in both cases. The relative reduction in test MRR for seed s is calculated as

$$\Delta_{m,s}^{\text{rel}}(\mathcal{B}_{b,q}) = 100 \cdot \frac{m_s^{\text{clean}} - m_s^{b,q}}{m_s^{\text{clean}}} \tag{6}$$

For each quantile, we average the relative MRR reduction across the set of random seeds $\mathcal{S}$:

$$\overline{\Delta}_m^{\text{rel}}(\mathcal{B}_{b,q}) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \Delta_{m,s}^{\text{rel}}(\mathcal{B}_{b,q}) \tag{7}$$

Finally, for each dataset–model–budget setting, we report the largest average degradation observed across the score quantiles:

$$\Delta_m^{\text{score}}(b) = \max_{q \in \mathcal{Q}} \overline{\Delta}_m^{\text{rel}}(\mathcal{B}_{b,q}) \tag{8}$$

For comparison, let $\mathcal{B}_b^{\text{base}}$ denote the subset of b training triples perturbed by the baseline strategy. Its relative effect under the same perturbation budget is computed as:

$$\Delta_m^{\text{base}}(b) = 100 \cdot \frac{m(\theta_{\text{clean}}; \mathcal{G}_{\text{test}}) - m(\theta_{\mathcal{B}_b^{\text{base}}}; \mathcal{G}_{\text{test}})}{m(\theta_{\text{clean}}; \mathcal{G}_{\text{test}})} \tag{9}$$

These metrics allow us to examine whether selecting triples from particular score regions results in a larger relative MRR degradation than the baseline perturbation strategy under the same budget. Table 2 reports the test MRR drops under the baseline and our proposed score-quantile strategy. The color indicates the score quantile that yields the largest drop for each dataset–model–budget setting. The table shows that for the denser datasets, larger drops tend to occur when perturbations target mid-high to high score quantiles. In contrast, for

DB	Ratio	Att.	TransH	DistMult	ComplEx	DualE	QMult	Keci	RotatE	MuRE
KINSHIP	0.02	Base	-3.2%	-2.7%	-2.8%	-3.5%	-2.9%	-4.9%	-1.7%	-4.7%
		Surr	**-10.5%**	**-5.0%**	**-4.7%**	**-4.9%**	**-5.1%**	**-5.9%**	**-4.9%**	**-5.2%**
	0.04	Base	-5.4%	-5.5%	-6.2%	-7.8%	-6.7%	-9.9%	-3.9%	-6.2%
		Surr	**-15.9%**	**-8.2%**	**-8.3%**	**-9.8%**	**-11.6%**	**-10.0%**	**-6.5%**	**-9.0%**
	0.06	Base	-5.9%	-8.4%	-9.8%	-11.1%	-10.1%	-11.7%	-6.5%	-9.3%
		Surr	**-20.6%**	**-13.3%**	**-12.5%**	**-13.1%**	**-14.1%**	**-13.3%**	**-8.5%**	**-11.8%**
UMLS	0.02	Base	-3.5%	-5.1%	-4.9%	-6.2%	-5.2%	-4.9%	-2.7%	-5.2%
		Surr	**-6.3%**	**-6.9%**	**-6.2%**	**-7.6%**	**-7.8%**	**-6.1%**	**-4.7%**	**-7.7%**
	0.04	Base	-6.0%	-10.4%	-10.3%	-11.4%	-10.5%	-8.0%	-6.1%	-9.8%
		Surr	**-9.2%**	**-12.8%**	**-12.7%**	**-13.2%**	**-13.2%**	**-11.9%**	**-5.0%**	**-13.4%**
	0.06	Base	-9.1%	-14.7%	-14.9%	-15.9%	-15.3%	-13.1%	-10.1%	-15.1%
		Surr	**-12.4%**	**-16.5%**	**-16.9%**	**-19.0%**	**-18.6%**	**-16.0%**	**-6.3%**	**-18.0%**
FB15k-237	0.02	Base	-1.9%	-4.1%	-3.2%	-2.3%	-1.0%	-4.4%	-1.2%	-2.3%
		Surr	**-3.4%**	**-5.8%**	**-4.0%**	**-5.7%**	**-3.6%**	**-12.9%**	**-2.9%**	**-4.8%**
	0.04	Base	-3.3%	-6.0%	-4.9%	-3.3%	-3.8%	-11.9%	-2.4%	-4.3%
		Surr	**-5.4%**	**-8.4%**	**-7.3%**	**-5.1%**	**-5.8%**	**-10.8%**	**-4.0%**	**-7.0%**
	0.06	Base	-4.4%	-8.2%	-6.6%	-5.5%	-5.1%	-9.3%	-4.5%	-7.0%
		Surr	**-6.3%**	**-12.4%**	**-11.8%**	**-8.8%**	**-7.7%**	**-11.8%**	**-4.9%**	**-9.9%**
NELL-995-h100	0.02	Base	-4.9%	-0.2%	-0.1%	-2.8%	-3.4%	-2.2%	-2.1%	-1.4%
		Surr	**-14.4%**	**-6.3%**	**-8.4%**	**-6.3%**	**-5.6%**	**-8.6%**	**-3.2%**	**-4.0%**
	0.04	Base	-7.9%	-5.3%	-5.4%	-2.3%	-4.4%	-2.3%	-5.0%	-4.1%
		Surr	**-11.8%**	**-9.9%**	**-11.2%**	**-6.8%**	**-12.9%**	**-13.3%**	**-5.2%**	**-7.5%**
	0.06	Base	-11.5%	-2.9%	-6.3%	-4.3%	-8.1%	-7.0%	-5.2%	-9.6%
		Surr	**-15.0%**	**-11.9%**	**-13.9%**	**-11.7%**	**-12.9%**	**-13.5%**	**-9.0%**	-9.3%
WN18RR	0.02	Base	-4.3%	-4.3%	-5.5%	-1.3%	-1.4%	-3.6%	-3.8%	-5.7%
		Surr	**-5.6%**	**-5.4%**	**-12.5%**	**-37.7%**	**-0.5%**	**-9.2%**	**-5.2%**	**-6.0%**
	0.04	Base	-8.6%	-7.9%	-11.7%	-18.2%	-5.8%	-6.2%	-5.0%	-9.6%
		Surr	**-8.9%**	**-9.1%**	**-16.8%**	**-32.4%**	**-9.2%**	**-10.4%**	**-10.6%**	**-12.3%**
	0.06	Base	-10.2%	-12.2%	-17.2%	-30.9%	-11.1%	-7.9%	-6.9%	-13.5%
		Surr	**-13.8%**	**-13.5%**	**-25.8%**	**-44.5%**	**-13.8%**	**-13.0%**	**-14.6%**	**-16.5%**

Quantiles: Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10

Table 2. Relative drops in test MRRs for the random baseline (Base) and surrogate-based attack (Surr) across datasets, perturbation ratios, and KGE models. For Surr, the colored square indicates the score quantile producing the reported drop: lighter magenta denotes lower quantiles, while darker magenta denotes higher quantiles.

the sparser datasets, the largest drops more often arise from perturbing low to mid-low score quantiles. This creates a profile for the dataset-model pairs that highlights their weaknesses.

6 Discussion

Our results show that returned plausibility scores expose more than a final link-prediction ranking. They allow the training triples to be ordered, making it possible to identify score regions with different levels of poisoning sensitivity. Below we look into some specific cases.

6.1 Score Quantiles and Their Influence

The purpose of partitioning the training triples into score quantiles is to avoid searching over the complete training graph without guidance. Indeed, if perturbing triples from different score regions leads to similar performance degradation, score-guided selection offers no advantage over random sampling. However, Figure 2 and Table 2 show that this is generally not the case. For many dataset–model–budget settings, the largest degradation is concentrated in particular regions of the score distribution. Hence, the score ordering provides a useful signal for narrowing the candidate set available to an attacker. Here, it is important to distinguish a triple's plausibility from its *influence*. In KGE training, observed graph triples serve as positive examples, while negative candidates are generated during training. The learning process encourages observed triples to receive higher scores than generated negative candidates. A triple's final score therefore reflects how well it fits the relational patterns encoded by the learned parameters. Its influence, however, depends on how much replacing the triple changes the model's rankings.

Triples in the lowest quantiles are only weakly captured by the KGE model, so replacing them may affect evidence on which the model relies only marginally. Triples in the middle-to-upper quantiles are more strongly encoded in the learned representation and may therefore contribute more to shaping the entity and relation embeddings. Perturbing them alters training evidence that has contributed to the learned embeddings and can affect multiple link-prediction queries. Yet, the highest-scoring triples represent a different case. Although a triple in Q_{10} is highly compatible with the learned representation, its entities, relation, or relational pattern may be strongly supported by many other training triples. The remaining evidence can therefore compensate for its perturbation. This helps explain why Q_{10} is not consistently the most damaging quantile and why middle-to-upper quantiles often provide a more effective combination of strong model compatibility and limited redundant support.

Therefore, a triple's plausibility score should not be interpreted as a direct measure of its influence. The score reflects the plausibility assigned to the triple by the learned model, whereas its influence depends on the amount of supporting evidence elsewhere in the training graph. Such support may arise from repeated relations, shared entities, alternative relational paths, or semantically similar patterns. For example, consider two highly scored triples. One involves frequent entities and a common relation, while the other contains an entity or relation supported by only a few training facts. Perturbing the first triple may have a limited effect because many other facts continue to constrain the same embeddings. Perturbing the second can alter a larger share of the available evidence and may therefore have a stronger effect, even if its plausibility score is lower. This helps explain why the highest score does not necessarily correspond to the greatest poisoning effect. A systematic characterization of these forms of support requires a dedicated structural and semantic analysis of the knowledge graph.

6.2 Effect of Dataset Density

Differences in graph density help explain why the most effective score regions vary from one knowledge graph to another. As shown in Table 1, KINSHIP, UMLS, and FB15k-237 are substantially denser than NELL-995-h100 and WN18RR. Generally, in denser graphs, entities and relations occur in more training triples, which constrains their embeddings more strongly and provides repeated relational evidence. Accordingly, Figure 2 and Table 2 show that effective perturbations in KINSHIP, UMLS, and FB15k-237 occur mainly in the middle-to-upper score regions. Triples in these quantiles are sufficiently well represented to affect the learned rankings, while potentially receiving less redundant support than many triples in Q_{10}. This is consistent with the tendency of perturbations in $Q_7 - Q_9$ to produce larger degradations than those in Q_{10} in several dense-dataset settings. However, this pattern changes in sparse datasets.

In NELL-995-h100 and WN18RR entities and relations occur in fewer training triples; therefore, parts of the embedding vectors remain weakly constrained. Triples in the low-to-middle score range, $Q_1 - Q_5$, may involve rare entities or infrequent relational patterns supported by a few training facts. Although these triples receive relatively low scores, each may constitute a meaningful part of the limited evidence available for the corresponding entities and relations. Perturbing such a triple therefore removes one of the few available training facts and introduces a corrupted one. Because few alternative facts remain to constrain the embeddings, the perturbation can change their representations. A low plausibility score therefore does not necessarily imply limited influence in sparsely connected graphs. This aligns with the stronger impact of lower score quantiles in Figure 2 and the large relative MRR degradations reported for NELL-995-h100 and WN18RR in Table 2. Dataset density therefore affects not only the magnitude of poisoning damage, but also where the most effective attack candidates occur in the score distribution.

6.3 Effect of KGE Approaches

Dataset density does not fully determine the vulnerable score region. As shown in Figure 2 and Table 2 models trained on the same dataset can show variations in their degradation patterns which suggests that their response to perturbation also depends on the model's geometry and training. Here, model geometry refers to the structural constraints imposed by a KGE scoring function on how heads, relations, and tails interact. A poisoned triple therefore produces different parameter updates across models. In translational models, it can shift embeddings to satisfy an incorrect displacement, while in bilinear models it changes shared coordinate-wise interactions. Because the same entity and relation embeddings are reused across many triples, these updates can propagate beyond the perturbed fact. Model geometry thus affects how strongly a local perturbation spreads through the score function and alters other link predictions. For instance DualE shows substantially larger score-based degradation than several

other models on WN18RR and reaches a relative MRR drop of 44.5% at a budget of (0.06). Overall, our results show that an attacker can use the plausibility scores returned by the model to identify where poisoning is most effective.

7 Conclusion and Future work

Compared to prior white-box poisoning methods that require access to model parameters and gradients, our approach operates with a more restricted access, relying only on model outputs. This perspective shows that the performance of link prediction algorithms can be compromised by editing a small set of training triples that are selected based on the plausibility scores. It also allows us to identify which parts of the score distribution are most influential for different datasets and models, and therefore where poisoning is most effective. In our current setup, after selecting a triple from a score quantile, we perturb it by randomly substituting its head, relation, or tail. Future work can replace this random replacement with score-guided edits. For example, an attacker can delete a selected training triple and inject an alternative false triple that receives a high score. This allows the added triple to appear plausible, despite being incorrect. This perspective motivates the use of returned confidence scores as an attack signal. A service that exposes continuous scores reveals the model's *graded* belief over candidates. These scores can therefore be used to identify which training triples the model strongly accepts, which ones it strongly rejects. Perturbing these regions allows us to test whether the model's downstream link-prediction decisions depend primarily on strongly accepted facts, already rejected facts, or weakly supported facts.

Acknowledgment: This work has been supported by the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) within the project SAIL (grant no NW21-059D) and the project WHALE (LFN 1-04) funded under the Lamarr Fellow Network programme.

Reproducibility: We made all code and accompanying resources publicly available at `https://doi.org/10.5281/zenodo.20351857`. This includes the scripts for generating score-based perturbed datasets, running experiments and evaluations, performing statistical tests, and a README.txt file. The supplementary material also includes a PDF with boxplot summaries of the performance across all 9,792 experiments.

Competing interests: The authors have no relevant financial or non-financial interests to disclose.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: Dbpedia: A nucleus for a web of open data. In: The Semantic Web, 6th International Semantic Web Conference ISWC (2007)
2. Balazevic, I., Allen, C., Hospedales, T.: Multi-relational poincaré graph embeddings. Advances in neural information processing systems **32** (2019)

3. Banerjee, P., Chu, L., Zhang, Y., Lakshmanan, L.V., Wang, L.: Stealthy targeted data poisoning attack on knowledge graphs. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). pp. 2069–2074. IEEE (2021)
4. Betz, P., Meilicke, C., Stuckenschmidt, H.: Adversarial explanations for knowledge graph embeddings. In: IJCAI. vol. 2022, pp. 2820–2826 (2022)
5. Bhardwaj, P., Kelleher, J.D., Costabello, L., O'Sullivan, D.: Adversarial attacks on knowledge graph embeddings via instance attribution methods. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. pp. 8225–8239. Association for Computational Linguistics (2021). https://doi.org/10.18653/V1/2021.EMNLP-MAIN.648, https://doi.org/10.18653/v1/2021.emnlp-main.648
6. Bhardwaj, P., Kelleher, J.D., Costabello, L., O'Sullivan, D.: Poisoning knowledge graph embeddings via relation inference patterns. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP (2021)
7. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250 (2008)
8. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Advances in neural information processing systems **26** (2013)
9. Cao, Z., Xu, Q., Yang, Z., Cao, X., Huang, Q.: Dual quaternion knowledge graph embeddings. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 6894–6902 (2021)
10. Dalton, J., Dietz, L., Allan, J.: Entity query feature expansion using knowledge base links. In: The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (2014)
11. Demir, C., Moussallem, D., Heindorf, S., Ngomo, A.C.N.: Convolutional hypercomplex embeddings for link prediction. In: Asian Conference on Machine Learning. pp. 656–671. PMLR (2021)
12. Demir, C., Ngomo, A.C.: Clifford embeddings–a generalized approach for embedding in normed algebras. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases (2023)
13. Denham, W.W.: The detection of patterns in Alyawara nonverbal behavior. Ph.D. thesis, University of Washington. (1973)
14. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
15. Duddu, V., Boutet, A., Shejwalkar, V.: Quantifying privacy leakage in graph embedding. In: MobiQuitous 2020-17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services. pp. 76–85 (2020)
16. Ferrucci, D.A., Brown, E.W., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J.M., Schlaefer, N., Welty, C.A.: Building watson: An overview of the deepqa project. AI Mag. **31**(3), 59–79 (2010)
17. Heindorf, S., Potthast, M., Stein, B., Engels, G.: Vandalism detection in wikidata. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 327–336 (2016)
18. Heindorf, S., Scholten, Y., Engels, G., Potthast, M.: Debiasing vandalism detection models at wikidata. In: The World Wide Web Conference. pp. 670–680 (2019)

19. Kapoor, S., Sharma, A., Röder, M., Demir, C., Ngomo, A.N.: Performance evaluation of knowledge graph embedding approaches under non-adversarial attacks. CoRR **abs/2407.06855** (2024)
20. Kapoor, S., Sharma, A., Röder, M., Demir, C., Ngomo, A.N.: Robustness evaluation of knowledge graph embedding models under non-targeted attacks. In: Curry, E., Acosta, M., Poveda-Villalón, M., van Erp, M., Ojo, A.K., Hose, K., Shimizu, C., Lisena, P. (eds.) The Semantic Web - 22nd European Semantic Web Conference, ESWC 2025, Portoroz, Slovenia, June 1-5, 2025, Proceedings, Part I. pp. 264–281. Lecture Notes in Computer Science, Springer (2025). https://doi.org/10.1007/978-3-031-94575-5_15, https://doi.org/10.1007/978-3-031-94575-5_15
21. Liu, C., Wang, Y., Cao, H., Liu, B., Jiang, D., Xu, L.: Non-targeted adversarial attacks on vision-language models via maximizing information entropy (2024)
22. McCray, A.T.: An upper-level ontology for the biomedical domain. Comparative and Functional genomics **4**(1), 80–84 (2003)
23. Memariani, A., Röder, M., Sharma, A., Demir, C., Ngomo, A.C.N.: Link prediction under non-targeted attacks: Do soft labels always help? In: International Semantic Web Conference. pp. 99–121. Springer (2025)
24. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
25. Nickel, M., Tresp, V., Kriegel, H.P., et al.: A three-way model for collective learning on multi-relational data. In: Icml. vol. 11, pp. 3104482–3104584 (2011)
26. Pezeshkpour, P., Tian, Y., Singh, S.: Investigating robustness and interpretability of link prediction via adversarial modifications. In: 1st Conference on Automated Knowledge Base Construction, AKBC 2019, Amherst, MA, USA, May 20-22, 2019 (2019), https://openreview.net/forum?id=Hkg7rbcp67
27. Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.: Knowledge graph embedding for link prediction: A comparative analysis. ACM Trans. Knowl. Discov. Data **15**(2), 14:1–14:49 (2021)
28. Sarabadani, A., Halfaker, A., Taraborelli, D.: Building automated vandalism detection tools for wikidata. In: Proceedings of the 26th International Conference on World Wide Web Companion. pp. 1647–1654 (2017)
29. Sharma, A., Kouagou, N., Ngomo, A.C.N.: Resilience in knowledge graph embeddings (2024)
30. Sharma, A., Kouagou, N.J., Ngomo, A.N.: Resilience in knowledge graph embeddings. Trans. Graph Data Knowl. **3**(2), 1:1–1:38 (2025). https://doi.org/10.4230/TGDK.3.2.1, https://doi.org/10.4230/TGDK.3.2.1
31. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space (2019)
32. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd workshop on continuous vector space models and their compositionality. pp. 57–66 (2015)
33. Trouillon, T., Dance, C.R., Gaussier, É., Welbl, J., Riedel, S., Bouchard, G.: Knowledge graph completion via complex tensor factorization. Journal of Machine Learning Research **18**(130), 1–38 (2017)
34. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International conference on machine learning. pp. 2071–2080. PMLR (2016)
35. Wang, Y., Huang, L., Yu, P.S., Sun, L.: Membership inference attacks on knowledge graphs. arXiv preprint arXiv:2104.08273 (2021)

36. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI conference on artificial intelligence. vol. 28 (2014)
37. Xiong, W., Hoang, T., Wang, W.Y.: Deeppath: A reinforcement learning method for knowledge graph reasoning (2017)
38. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases (2014)
39. Yang, G., Zhang, L., Liu, Y., Xie, H., Mao, Z.: Exploiting pre-trained language models for black-box attack against knowledge graph embeddings. ACM Transactions on Knowledge Discovery from Data **19**(1), 1–14 (2024)
40. You, X., Sheng, B., Ding, D., Zhang, M., Pan, X., Yang, M., Feng, F.: Mass: Model-agnostic, semantic and stealthy data poisoning attack on knowledge graph embedding. In: Ding, Y., Tang, J., Sequeda, J.F., Aroyo, L., Castillo, C., Houben, G. (eds.) Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023. pp. 2000–2010. ACM (2023). https://doi.org/10.1145/3543507.3583203, https://doi.org/10.1145/3543507.3583203
41. Zhang, H., Zheng, T., Gao, J., Miao, C., Su, L., Li, Y., Ren, K.: Data poisoning attack against knowledge graph embedding. In: Kraus, S. (ed.) Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019. pp. 4853–4859. ijcai.org (2019)
42. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. Advances in neural information processing systems **32** (2019)
43. Zhao, T., Chen, J., Ru, Y., Lin, Q., Geng, Y., Liu, J.: Untargeted adversarial attack on knowledge graph embeddings. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024. pp. 1701–1711. ACM (2024)