Mind The Margins: Probing Link Prediction via Perturbations

Adel Memariani
Data Science Group (DICE), Heinz Nixdorf Institute, Department of Computer Science, Paderborn University
Paderborn, Germany
adel.memariani@upb.de

Arnab Sharma
Data Science Group (DICE), Heinz Nixdorf Institute, Department of Computer Science, Paderborn University
Paderborn, Germany
arnab.sharma@upb.de

Michael Röder
Data Science Group (DICE), Heinz Nixdorf Institute, Department of Computer Science, Paderborn University
Paderborn, Germany
michael.roeder@upb.de

Axel-Cyrille Ngonga Ngomo
Data Science Group (DICE), Heinz Nixdorf Institute, Department of Computer Science, Paderborn University
Paderborn, Germany
axel.ngonga@upb.de

Abstract

Adversarial attacks pose a major threat to link prediction models. Carefully chosen triple-level perturbations can exploit weaknesses in the learned representations and change the model's ranking decisions. Despite their importance, adversarial attacks on knowledge graph embeddings (KGE) models remain less studied than their counterparts in areas such as image classification. One reason is that KGE inputs are discrete relational triples, not continuous pixel values. Moreover, KGE scores are computed within multi-relational prediction settings, where each relation can produce a different score distribution. As a result, score-based attack strategies cannot be directly transferred to KGE models. Studying training-data perturbations therefore both exposes vulnerabilities and identifies informative graph edits that can improve performance. In this work, we use adversarial triple modifications as controlled probes of KGE decision behavior. We introduce a decision-margin perspective that measures how clearly each training triple is ranked above its strongest negative alternatives. Our results reveal systematic vulnerability patterns, as well as improvement opportunities, that are shared across different KGE models. At the same time, these effects are not determined by the embedding model alone. The impact of training data perturbations also depends on the structure of the underlying knowledge graph. Overall, our study reframes adversarial probes as diagnostic instruments for KGE models.

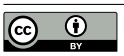
CCS Concepts

• **Computing methodologies** → *Learning to rank*.

Keywords

Link Prediction; Knowledge Graph Embeddings

ACM Reference Format:
Adel Memariani, Arnab Sharma, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2026. Mind The Margins: Probing Link Prediction via Perturbations. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 07–11, 2026, Rome, Italy*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3799682.3840996



CIKM '26, Rome, Italy

ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3840996

1 Introduction

Link-prediction models are often trained on knowledge graphs that are derived from large, evolving, and publicly editable sources such as DBpedia [2] and Wikidata [30]. While this supports broad coverage, it also increases the risk that incorrect, misleading, or malicious facts enter the knowledge graph through extraction errors, vandalism, or adversarial manipulation [5, 15, 20, 25]. Mainly, predicting missing links in KGs is time-intensive due to their discrete and symbolic structure [4, 5, 35, 38].

Knowledge graph embedding (KGE) models learn representations of entities and relations and use model-specific scoring functions to rank candidate triples for link prediction [8, 12, 13]. Because these scores are learned from training triples, corrupted facts can change how the model ranks candidate facts. Therefore, the reliability of such models depends on the quality of the training graph. Although, KGE models differ in their scoring functions and representational biases, however they are all trained to distinguish valid relational facts from implausible candidates. We therefore study how controlled additions and deletions of triples expose systematic vulnerabilities in KGE models. A central question is whether these vulnerabilities are mainly caused by specific scoring functions or whether they reflect a broader weakness in the link-prediction setting. We examine the latter possibility. If diverse KGE models exhibit similar vulnerability patterns, then these effects are not solely attributable to the design of individual scoring functions. Instead, they point to weaknesses arising from the shared ranking objective that KGE models are trained to optimize. In this paper, we introduce a separation-margin viewpoint for probing systematic vulnerabilities in link-prediction systems.

Existing attacks on KGE models are usually framed either as targeted or non-targeted attacks. Targeted attacks aim to change the prediction for a specific query or target triple [4–6, 35], while non-targeted attacks aim to degrade the overall link-prediction performance of the model [17, 38]. Although both settings can be used to evaluate different aspects of the robustness of KGE models, they provide limited insight into which types of training triples are most influential for the learned link-prediction function. For instance, targeted attacks reveal how to affect individual predictions; however, they do not necessarily explain which training triples are broadly influential for the learned representation. Non-targeted attacks, on the other hand, quantify global degradation and often use the selected perturbations primarily to reduce metrics such as Mean Reciprocal Rank (MRR). Consequently, they provide limited

insight into what distinguishes more influential training triples from less influential ones. In this work, we use adversarial triple modifications as controlled probes of KGE decision behavior. Our central idea is a *separation-margin* perspective, i.e., we characterize each training triple by how well it is separated from its strongest negative samples. This allows us to distinguish (i) high-confidence facts that the model strongly supports, (ii) borderline facts close to the decision boundary, and (iii) under-supported facts that the model fails to separate from competing unobserved triples. These categories provide a model-agnostic way to study vulnerabilities because they are defined by the link-prediction task rather than a specific scoring function. Specifically, we shift the focus from how to attack a KGE model to which triples in knowledge graphs are most influential for link-prediction performance. We use separation-margin-guided edits as a diagnostic tool. Therefore, our work is positioned between targeted and non-targeted poisoning attacks. Unlike targeted attacks, we do not focus only on changing the rank of one predefined fact. Unlike standard non-targeted attacks, we do not treat global performance degradation as the only outcome of interest. Instead, we use poisoning-style triple edits to study the attack relevance of individual training triples. In our setting, a triple is considered attack-relevant if its removal substantially changes the model's ranking performance. Similarly, a triple that is not present in the knowledge graph is considered attack-relevant if its addition causes a substantial change in ranking performance. Such changes may correspond to either an improvement or a degradation in performance. Specifically, unlike existing work that has primarily focused on the question, *How can we successfully attack a KGE model?* we focus on the more fundamental question, *Which triples are influential for predictions, and what distinguishes them?* This perspective clarifies the role of adversarial attacks in our study. We use poisoning attacks not merely as a threat model, rather as a tool for measuring the attack relevance of training triples. Thus, adversarial edits become controlled interventions for probing the importance of training triples within the link-prediction task.

2 Related Work

Adversarial robustness of KGE models has mainly been studied through training-time data poisoning, where an attacker modifies the training graph to change the model's predictions. Early work by Zhang et al. [36] showed that small additions or deletions of facts can manipulate the plausibility of selected target triples. Their work also highlights a central challenge for adversarial learning on knowledge graphs: attacks developed for conventional graph data or continuous input domains do not directly transfer to KGs, since KGs are symbolic, typed, and multi-relational. Subsequent work by Bhardwaj et al. [5, 6] used instance attribution to identify training triples that are influential for particular target predictions. They estimate influence through instance similarity, gradient similarity, and influence functions, and then use the selected triples for adversarial deletion or addition. Betz et al. [4] further connect adversarial attacks with interpretability by using rule learning and abductive reasoning to identify symbolic explanations for KGE predictions. In their view, harmful deletions can reveal graph-level evidence supporting a prediction.

More recently, Yang et al. [34] exploit textual information associated with knowledge graph entities. They encode triples using pre-trained language models (PLM) and select influential neighboring triples in the PLM embedding space for targeted deletion or addition. They further incorporate structural information by fine-tuning the PLM on random-walk sequences over the KG. A related line of work focuses on noisy training data. For instance, Zhang et al. [37] propose a multi-task reinforcement learning framework that learns policy-based agents to identify and filter noisy triples before training KGE models. Sharma et al. [26] proposed resilience, where robustness, generalization, performance consistency, distribution adaptation, and missing-entry handling are treated as complementary aspects of reliable KGE models [26]. Note that the existing adversarial KGE studies mainly construct effective attacks against selected *target* triples. They often rely on auxiliary information, such as gradients, influence estimates, learned rules, abductive explanations, or textual embeddings. In contrast, we use graph edits as a diagnostic tool to understand how they affect overall link-prediction performance.

Black-box adversarial attacks have been widely studied in image classification, where the attacker cannot access model parameters or gradients but can query the model for predictions. Essentially, depending on the available feedback, prior works distinguish between decision-based attacks, which observe only the predicted label, and score-based attacks, which additionally exploit confidence or probability-like outputs, for instance, zeroth-order optimization attacks such as ZOO [7], query-efficient attacks based on natural evolution strategies [16], simple score-based attacks such as SimBA [14], random-search methods such as Square Attack [1], and evolutionary or genetic search strategies [23]. Note that these methods build on the fact that image inputs are continuous. Specifically, an attacker can iteratively apply small perturbations to pixel values and use model scores to navigate the local decision landscape. Importantly, this formulation does not transfer naturally to knowledge graph link prediction. A triple is a symbolic relational fact. Modifying a triple usually means replacing an entity or relation, adding a new fact, or deleting an existing one. Such operations are discrete and may change the semantics of the graph rather than producing a small perturbation of the same input. Moreover, KGE models are evaluated through ranking, i.e., the effect of a perturbation is determined not only by whether one label changes, but also by how the score of a positive triple shifts relative to the scores of many competing candidate triples. Therefore, we cannot simply take the approaches from the domain of image classification and apply them to KGE models. In our proposed method, we rely only on KGE plausibility scores to compute separation margins between observed triples and competing candidates. As shown in the following sections, these margins distinguish between triples that are well separated, near the hard-negative boundary, or weakly separated.

3 Methodology

A direct way to assess a triple's importance for a KGE model is to edit the knowledge graph, retrain the model, and compare the resulting link prediction performance with the model trained before the edits. This procedure, however, does not scale: for large knowledge graphs, re-training and re-evaluation across many possible

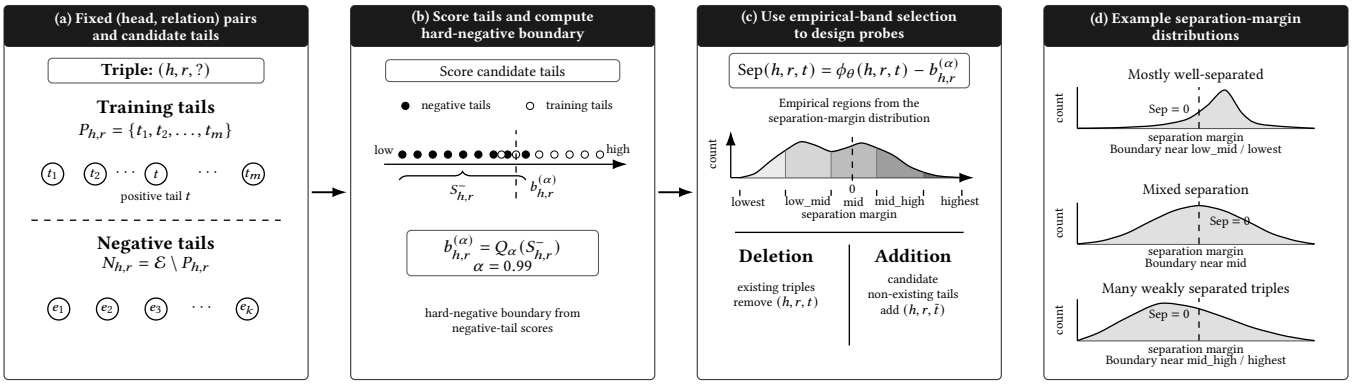


Figure 1: Separation-margin probes. For each (h,r,?), tail scores define a hard-negative boundary and empirical regions for deletion/addition. Box (d) shows that Sep=0 can fall in different regions across dataset–model combinations.

deletions or additions quickly becomes computationally infeasible. The separation margin provides a lightweight and model-agnostic way to quantify how well a training fact is captured by a KGE-based link prediction system. Rather than relying on costly retraining, we estimate a triple's importance directly from the model's link prediction scores. We define a separation margin that quantifies the extent to which a training triple is separated from competitive negative alternatives. Importantly, the margin should not be interpreted as a semantic truth indicator; rather, it measures how strongly the trained link prediction system supports the fact relative to its own ranking method. Moreover, in Section 3.5, we consider a more restrictive scenario in which access to the target model's link prediction scores is unavailable. In this case, we estimate separation margins using an ensemble of surrogate models. The effectiveness of this ensemble-based estimate shows that the proposed margin is not limited to settings with score access. It can also characterize which facts are likely to be well captured or overlooked by link prediction systems under substantially reduced access assumptions, requiring only a sufficiently large subset of triples for margin estimation. In particular, we characterize training-triple influence by measuring how well each triple is separated from its hardest negative samples. Training triples may differ in influence because some express relational patterns that a given KGE model captures more easily. Our separation margin is computed for each fixed pair (h, r), rather than globally across all triples. This is important because different relations have different candidate entities and different score distributions, so raw scores are not directly comparable across relations. By fixing (h, r), we compare a training tail only against alternative tails for the same query $(h, r, ?)$. This matches the ranking setting used in link prediction.

3.1 Separation Margin

Raw triple scores are not directly comparable across relations as each $(h, r, *)$ query has its own score distribution. Therefore, we do not use the score of a training triple as an absolute quantity. Instead, we measure how well the training triple is separated from hard competing negative tails within the same tail-prediction problem. Let $\mathcal{G}_{\text{train}} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ denote the training graph, where $\mathcal{E}$ is the entity set and $\mathcal{R}$ is the relation set. A trained KGE model with parameters θ defines a scoring function $\phi_\theta : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R}$, where larger scores indicate higher model plausibility. For a training triple $\tau = (h, r, t)$, we fix the pair (h, r) and consider the corresponding tail-prediction problem $(h, r, ?)$. The set of training tails for this fixed pair is: $P_{h,r} = \{\, t' \in \mathcal{E} \mid (h, r, t') \in \mathcal{G}_{\text{train}} \,\}$. All remaining entities are treated as candidate competing tails: $N_{h,r} = \mathcal{E} \setminus P_{h,r}$. The model scores every candidate tail for the fixed pair (h, r): $S_{h,r} = \{\phi_\theta(h, r, e) \mid e \in \mathcal{E}\}$. We then separate the scores of the training tails from the scores of the candidate competing tails:

$$S^+_{h,r} = \{\phi_\theta(h, r, t') \mid t' \in P_{h,r}\},\ S^-_{h,r} = \{\phi_\theta(h, r, e) \mid e \in N_{h,r}\}. \quad (1)$$

The central quantity in our approach is the hard-negative boundary. Instead of comparing a training triple to the average negative tail, we compare it to the upper tail of the negative-score distribution. Let α be a high quantile, set to $\alpha = 0.99$ in our experiments. The hard-negative boundary for (h, r) is defined as

$$b^{(\alpha)}_{h,r} = Q_\alpha\left(S^-_{h,r}\right), \quad (2)$$

where Q_α denotes the empirical α-quantile. This boundary represents the score level reached by the strongest competing tails for the same prediction problem. Importantly, $b^{(\alpha)}_{h,r}$ is not a global constant and is not assumed to be zero. It depends on the relation, the head entity, and the score distribution obtained from a trained model for the specific completion task $(h, r, ?)$. The separation margin of a training triple (h, r, t) is then defined as:

$$\mathrm{Sep}(h, r, t) = \phi_\theta(h, r, t) - b^{(\alpha)}_{h,r}. \quad (3)$$

The sign of the margin has an intuitive ranking interpretation. If $\mathrm{Sep}(h, r, t) > 0$, then the training tail is scored above the hard-negative boundary. If $\mathrm{Sep}(h, r, t) \approx 0$, then the training triple lies close to the boundary. If $\mathrm{Sep}(h, r, t) < 0$, then the training tail is scored below the hard-negative boundary. Therefore, the margin measures whether a training triple is safely separated from the strongest competing tails in its own prediction context. After computing $\mathrm{Sep}(h, r, t)$ for all training triples, we sort the triples in ascending order of separation score.

Let $\mathcal{T} = [\tau_{(1)}, \tau_{(2)}, \ldots, \tau_{(m)}]$ be the sorted list, where $\mathrm{Sep}(\tau_{(1)}) \leq \mathrm{Sep}(\tau_{(2)}) \leq \cdots \leq \mathrm{Sep}(\tau_{(m)})$ and $m = |\mathcal{G}_{\text{train}}|$. In our method, we use this ordered margin distribution to construct the regions considered

for perturbation. Triples at the beginning of the ordering have the lowest margins, whereas triples at the end have the highest margins. For a perturbation ratio ρ, the edit budget is: $B(\rho) = \text{round}\left(\rho \cdot |\mathcal{G}_{\text{train}}|\right)$. The lowest-margin deletion set is: $\mathcal{D}_{\text{lowest}} = \{\tau_{(1)}, \ldots, \tau_{(B)}\}$, and the highest-margin deletion set is: $\mathcal{D}_{\text{highest}} = \{\tau_{(m-B+1)}, \ldots, \tau_{(m)}\}$. We also define empirical interior regions over the same ordered margin distribution. For a band with fractional lower and upper edges $0 \le a < b \le 1$, we convert these fractions into index cutoffs in the sorted list. The corresponding empirical region is:

$$\mathcal{R}_{[a,b)} = \left\{\tau_{(i)} \;\middle|\; \lfloor a\, m \rfloor + 1 \le i \le \lfloor b\, m \rfloor\right\} \tag{4}$$

where $m = |\mathcal{G}_{\text{train}}|$. In our experiments, we refer to the interior regions as: Low-Mid = $[0.20, 0.40)$, Mid = $[0.40, 0.60)$, Mid-High = $[0.60, 0.80)$. Together with the Lowest, and Highest regions, these regions allow us to perturb different parts of the learned rankings while keeping the budget fixed.

3.2 Generation of perturbed datasets

Let $\mathcal{G}_{\text{train}} = \{x_1, \ldots, x_n\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ denote the reference training graph, where each $x_i = (h_i, r_i, t_i)$ has a separation score $\text{Sep}(x_i) \in \mathbb{R}$. To assess the effect of different perturbation levels, we define the edit budget as a ratio of the total training size of each dataset (see Equation 4). Using this budget, we construct edited training graphs through deletion-only or addition-only perturbations. The edits are guided by the separation margins. Deletion probes the effect of removing existing triples with different separation profiles, whereas addition probes the effect of injecting non-existing but separation-informed candidate triples. Primarily, we categorize training triples according to their separation margin relative to the hard-negative boundary. High-separation triples are those for which the model assigns the observed training tail a score clearly above the hard-negative boundary for the same (h, r) query. These triples can therefore be interpreted as facts that the model has learned to distinguish from competitive negative alternatives. In contrast, low-separation triples are facts whose scores remain close to, or below, the hard-negative region, indicating that the model does not clearly distinguish them from negative candidates. The intermediate bands (Low-Mid, Mid, and Mid-High) cover triples that are neither weakly separated nor strongly separated from competing negatives. Including these bands allows us to compare the effect of perturbing triples across different levels of separation.

3.3 Deletion-only dataset generation

For deletion-only perturbations, a policy p selects exactly $B(\rho)$ training triples for removal. The resulting edited graph is:

$$\mathcal{G}^{(\rho)}_{\text{del},p} = \mathcal{G}_{\text{train}} \setminus D^{(\rho)}_p,\ |D^{(\rho)}_p| = B(\rho).$$

Let $x_{(1)}, \ldots, x_{(n)}$ denote the training triples sorted in ascending order of their separation margins, i.e., $\text{Sep}(x_{(1)}) \le \cdots \le \text{Sep}(x_{(n)})$. We define five deletion policies over this ordering. The Lowest policy removes the first $B(\rho)$ triples, corresponding to the least separated training facts. The Highest policy removes the last $B(\rho)$ triples, corresponding to the most strongly separated facts. The remaining three policies target the interior of the separation distribution: Low-Mid, Mid, and Mid-High select triples from the empirical rank intervals $[0.2, 0.4)$, $[0.4, 0.6)$, and $[0.6, 0.8)$, respectively. Within each interior interval, we do not select only the lowest- or highest-ranked triples. Instead, we select $B(\rho)$ triples at evenly spaced positions across the interval. This gives a representative subset of each interior region: Low-Mid samples the lower-middle part of the separation distribution, Mid samples the central part, and Mid-High samples the upper-middle part. Together with Lowest and Highest, these policies therefore perturb five ordered regions of the separation distribution, from least separated to most strongly separated training facts.

3.4 Addition-only dataset generation

For addition-only perturbations, we augment the training graph with non-existing triples obtained by replacing the tail of a training triple with a negative-tail candidate. For each training triple $x = (h, r, t) \in \mathcal{G}_{\text{train}}$, the separation computation stores a candidate tail for each policy: $\pi \in \{\text{Lowest, Low-Mid, Mid, Mid-High, Highest}\}$. The Lowest and Highest candidates correspond to the absolute lowest and highest candidate margins for the same (h, r) query, respectively. The Low-Mid, Mid, and Mid-High candidates are drawn from progressively higher interior empirical regions of the query-local negative-tail separation-margin distribution. For a policy π, the candidate generated from $x = (h, r, t)$ is $(h, r, \tilde{t}_\pi(x))$. We accept this candidate only if it is not already in the original training graph and has not already been selected for the same addition dataset. For each perturbation ratio ρ, we uniformly sample source triples from the training graph and add policy-specific candidates $(h, r, \tilde{t}_\pi(x))$ until the budget $B(\rho)$ is reached. Denoting the selected additions by $A^{(\rho)}_\pi$, the edited training graph is $\mathcal{G}^{(\rho)}_{\text{add},\pi} = \mathcal{G}_{\text{train}} \cup A^{(\rho)}_\pi,\ |A^{(\rho)}_\pi| = B(\rho)$. This design allows us to examine how link-prediction performance changes when we add previously unobserved triples with different levels of separation from the hard-negative boundary.

3.5 Ensemble-Based Separation

The separation policies described above select perturbations using the scores of the model under analysis. We additionally consider a reduced-access setting in which the target model's scores are unavailable. In this setting, perturbations are selected using an ensemble of proxy KGE models rather than the target model itself. Let $\mathcal{M} = \{M_1, \ldots, M_K\}$ denote the proxy ensemble. For each proxy model M_m, we compute the separation margin of each training triple using the same definition as before. Because different KGE models may use different score scales, we do not average raw separation margins. Instead, for each proxy model, we convert the margins into percentile ranks over the relevant selection set $\mathcal{X}$. For deletion, the selection set is the set of training triples, $\mathcal{X} = \mathcal{G}_{\text{train}}$. For addition, the selection set is the shared candidate pool, $\mathcal{X} = C$, where C is the union of candidates proposed by the proxy models. For each $x \in \mathcal{X}$, let $\rho_m(x) = \text{rank}_{\text{pct}}\big(s_m(x); \{s_m(x') : x' \in \mathcal{X}\}\big)$ denote the percentile rank of the separation margin assigned to x by proxy model M_m. The ensemble separation rank is then:

$$\bar{\rho}(x) = \frac{1}{|\mathcal{M}|} \sum_{M_m \in \mathcal{M}} \rho_m(x).$$

Low values of $\bar{\rho}(x)$ indicate items that are consistently weakly separated across the proxy models, while high values indicate items that are consistently strongly separated. We use $\bar{\rho}$ to define the same

five perturbation policies as in the single-model setting. The Lowest and Highest policies select items with the smallest and largest ensemble ranks, respectively. The Low-Mid, Mid, and Mid-High policies select representative items from the empirical rank intervals $[0.2, 0.4)$, $[0.4, 0.6)$, and $[0.6, 0.8)$. For deletion, the selected items are removed from $\mathcal{G}_{\text{train}}$; for addition, the selected candidates are added to $\mathcal{G}_{\text{train}}$. For ensemble-based addition, let C denote the set of candidate triples considered for addition. Each candidate $x = (h, r, e) \in C$ is evaluated by every proxy model M_m. Its separation margin under proxy M_m is $s_m(x) = \phi_m(h, r, e) - b_m(h, r)$, where $b_m(h, r)$ is the hard-negative boundary for query (h, r) under M_m. Since separation-margin scales differ across proxy models, we convert the margins produced by each proxy into percentile ranks over C. Let $\rho_m(x)$ denote the percentile rank of $s_m(x)$ under proxy M_m. For candidate x, we then compute the average percentile rank across the proxy models: $\bar{\rho}(x) = \frac{1}{|\mathcal{M}|} \sum_{M_m \in \mathcal{M}} \rho_m(x)$. The candidates are ordered according to $\bar{\rho}(x)$, yielding a consensus separation ordering across the proxy models. We then define the ensemble Lowest, Low-Mid, Mid, Mid-High, and Highest regions from this consensus ordering. This ensemble setting tests whether separation-margin-guided perturbations remain effective when the perturbation set is selected without access to the target model's scores. If the resulting perturbations still affect the target model, then the effect is less likely to be an artifact of a single scoring function and more likely to reflect a model-agnostic property of the link-prediction training objective.

4 Experimental Setup

Our experiments evaluate separation-margin-guided training-graph perturbations across seven benchmark datasets, six state-of-the-art KGE models, two perturbation types (*deletion* and *addition*), five perturbation ratios $\rho = 0.02, 0.04, 0.06, 0.08, 0.10$, and six selection variants per ratio: Random, Lowest, Low-Mid, Mid, Mid-High, and Highest. The Random baseline uses the same budget as the separation-based variants: for deletion, it removes uniformly sampled training triples; for addition, it samples feasible non-existing triples by preserving observed (h, r) and replacing the tail uniformly from the entity set. For each dataset–model combination, we also train one clean reference model on the original training graph, before any perturbations. The single-model setting therefore contains: $7_{\text{datasets}} \times 6_{\text{models}} \times 5_{\text{ratios}} \times 6_{\text{variants}} \times 2_{\text{types}} = 2{,}520$ perturbed configurations. Repeating the same design with ensemble-based selection gives another 2,520 configurations. Including the $7_{\text{datasets}} \times 6_{\text{models}} = 42$ clean reference configurations, the full study contains 2,520 + 2,520 + 42 = 5,082 distinct experiments.

4.1 Datasets

We use a set of benchmark knowledge graphs that cover different domains and structural properties. UMLS [19] contains biomedical concepts and relations from the Unified Medical Language System. KINSHIP [10] represents relations among members of the Alyawarra tribe. FB15k-237 [28] and CoDEx-S [24] are general-domain factual knowledge graphs. NELL-995-h100 [32] is derived from the Never-Ending Language Learning project and contains automatically extracted real-world facts. WN18RR [11] is derived from WordNet, a lexical knowledge base of word meanings and semantic relations [21]. For Nations dataset [18, 22], we use the benchmark splits distributed through KGDatasets [18] and PyKEEN [22]. The characteristics of the benchmark datasets are summarized in Table 1. For each dataset, we report the number of triples in the training, validation, and test splits, together with the number of unique entities $|\mathcal{E}|$ and relations $|\mathcal{R}|$. Let $\mathcal{T}$ denote the set of triples considered for computing the dataset statistics. We measure entity density as $\text{ED} = \frac{2|\mathcal{T}|}{|\mathcal{E}|}$, where the factor 2 accounts for the head and tail entity positions in each triple. Thus, ED measures the average number of entity occurrences per entity. Similarly, relation density is defined as $\text{RD} = \frac{|\mathcal{T}|}{|\mathcal{R}|}$, which measures the average number of triples per relation. To capture how evenly facts are distributed across entities and relations, we also report entropy-based statistics. Entity entropy is computed as:

$$\text{EE} = -\sum_{e \in \mathcal{E}} p_e \log_2 p_e, \qquad p_e = \frac{c_e}{\sum_{e' \in \mathcal{E}} c_{e'}}, \tag{5}$$

where c_e denotes the number of occurrences of entity e across the head and tail positions of all triples. Since each triple contributes one head and one tail entity occurrence, we have $\sum_{e' \in \mathcal{E}} c_{e'} = 2|\mathcal{T}|$.

Relation entropy is computed correspondingly as:

$$\text{RE} = -\sum_{r \in \mathcal{R}} p_r \log_2 p_r, \qquad p_r = \frac{c_r}{\sum_{r' \in \mathcal{R}} c_{r'}}, \tag{6}$$

where c_r denotes the number of triples containing relation r. Since each triple contains exactly one relation, we have $\sum_{r' \in \mathcal{R}} c_{r'} = |\mathcal{T}|$. Higher entropy indicates that facts are more evenly distributed across entities or relations, whereas lower entropy indicates concentration around a smaller subset of entities or relations.

Table 1: Dataset statistics: split sizes, density, and entropy.

Dataset	Train	Val	Test	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	ED	RD	EE	RE
Nations	1,592	199	201	14	55	284.57	36.22	3.72	5.20
KINSHIP	8,544	1,068	1,074	104	25	205.50	427.44	6.70	4.24
UMLS	5,216	652	661	135	46	96.73	141.94	6.60	4.47
FB15k-237	272,115	17,535	20,466	14,541	237	42.65	1308.51	12.72	6.55
CoDEx-S	32,888	1,827	1,828	2,034	42	35.93	870.07	10.20	3.40
NELL-995-h100	50,314	3,763	3,746	22,411	43	5.16	1344.72	12.01	4.19
WN18RR	86,835	3,034	3,134	40,943	11	4.54	8454.82	14.64	2.23

Table 1 shows that the datasets vary substantially in terms of entity and relation occurrence statistics. In terms of entity density (ED), Nations [18, 22] has the highest ED, followed by KINSHIP [10] and UMLS [19], while FB15k-237 [28] and CoDEx-S [24] are moderately dense, whereas NELL-995-h100 [32] and WN18RR [11] are sparse with respect to entity coverage. WN18RR [11] is particularly notable because it has the lowest ED but the highest RD. This means that individual entities appear in relatively few triples whereas each relation is used across many triples. Its low RE further indicates that relation occurrences are unevenly distributed across the relation set. Thus, WN18RR combines low average entity participation with concentrated relation usage.

4.2 Link Prediction Approaches

Our experiments include a diverse set of KGE models with different relational inductive biases. This allows us to test whether the proposed separation-margin analysis captures model-specific behavior

or a more general pattern shared across KGE models. TransH [31] is included because it directly addresses the limitations of simple translation models on mapping patterns such as one-to-many, many-to-one, and many-to-many relations. It projects entities onto relation-specific hyperplanes before applying translation, giving each entity a relation-dependent representation. DistMult [33] provides a multiplicative model based on diagonal bilinear interactions. Its symmetry constraint limits its ability to model antisymmetric relations. However, its relatively simple scoring function makes it suitable for examining how graph edits selected by our separation-margins affect link-prediction performance. ComplEx [29] extends DistMult [33] by using complex-valued embeddings. The complex-valued formulation distinguishes head and tail roles and better captures asymmetric relations. RotatE [27] represents relations as rotations in complex space and is designed to capture symmetry, antisymmetry, inversion, and composition. MuRE [3] applies relation-specific coordinate transformations together with translation-style matching. This gives it more flexibility than pure translation models for relation-dependent and high-arity patterns. Keci [9] uses Clifford-algebraic interactions, generalizing simpler multiplicative models such as DistMult [33] and ComplEx [29] and providing a more expressive algebraic scoring family. Together, these models cover projection-based, multiplicative, rotational, transformation-based, and algebraic KGE families.

5 Evaluation

Our evaluation asks whether the separation margin of a training triple is informative about its role in learning. We first characterize each training triple by its separation from candidate negative tails, and then test the effect of this characterization through controlled training-graph edits. To isolate the impact of each edit type, we evaluate deletion-only and addition-only settings separately. In both cases, separation-guided edits are compared against random baselines under the same perturbation budgets. This allows us to assess whether perturbing different empirical regions of the separation-margin distribution affects link-prediction performance differently from random edits. To compare performance changes across datasets and models with different reference MRR values, we report signed relative percentage changes with respect to the unperturbed reference model. The reference model is trained on the original training graph, before any deletion or addition perturbation. Let $\mathrm{MRR}^{\mathrm{ref}}_{d,m}$ denote the MRR of model m on dataset d without perturbations, and let $\mathrm{MRR}^{x}_{d,m,r}$ denote the MRR obtained after applying perturbation strategy x at ratio r. We define:

$$\Delta^{x}_{d,m,r}(\%) = 100 \cdot \frac{\mathrm{MRR}^{x}_{d,m,r} - \mathrm{MRR}^{\mathrm{ref}}_{d,m}}{\mathrm{MRR}^{\mathrm{ref}}_{d,m}}. \tag{7}$$

Negative values indicate MRR reductions relative to the reference model; positive values indicate relative improvements.

For brevity, we report the results as compact winner-summary tables. Rather than reporting every perturbation variant in full, each cell in Tables 2,4,3,5,7,6 show the strategy with the strongest effect for a given dataset–model–ratio combination (here, ratio means the budget for perturbation). Random perturbation provides a budget-matched baseline that does not use separation information. Therefore, differences between random and separation-guided edits reflect the effect of the selection criterion. All comparisons are centered on a single question: *Are separation-margin-guided edits more effective than random edits in changing link-prediction performance?* For deletion, this means identifying which strategy produces the largest MRR reduction relative to the unperturbed reference model. Unlike deletion, which removes existing training evidence for link prediction, addition expands the training graph with new triples. These additions may distort the learning signal or provide additional evidence that improves link-prediction performance. Therefore, addition can affect performance in either direction, and we report both the largest MRR reductions and the largest MRR improvements. Comparing random additions with separation-guided additions therefore allows us to test whether the separation margin provides a meaningful selection signal. We also observed occasional improvements in MRR after deletion. Such gains may occur when some training triples are less useful for learning patterns that generalize well to unseen triples. Removing such triples may therefore improve link-prediction performance. However, these gains are generally rare, small, and inconsistent across experimental settings. We therefore report and discuss deletion results mainly in terms of MRR reductions.

5.1 Deletions

The deletion results in Table 2 show that the strongest MRR reductions are not produced mainly by removing the absolute lowest- or highest-separation triples. Instead, the most damaging deletions often come from the interior regions of the empirical separation-margin distribution, especially Low-Mid, Mid, and Mid-High. This indicates that the model is not affected only by triples it fails to separate or by triples it has already separated with very high confidence. Rather, triples with intermediate separation appear to be more influential for link-prediction performance. These triples are sufficiently separated from competing negative tails but are not often among the highest-separation triples. Their removal therefore has a stronger effect on the resulting rankings. Moreover, separation-based regions produce the strongest effects in most cases, while random perturbation is more effective only rarely. This suggests that the perturbation effect is driven by where triples lie in the separation-margin distribution. In Section 7, we discuss why intermediate-separation triples can be more damaging to remove than the two extremes. We relate this behavior to the KGE training objective, where true triples must be ranked above hard competing candidates within their relation-specific prediction task.

5.2 Additions

Beyond deletion, the same separation-margin analysis allows us to study separation-informed additions to the training graph. For each training triple (h, r, t), we keep the query (h, r) fixed and replace the tail with a non-existing tail selected from a specific region of the separation-margin distribution of candidate tails for the same (h, r) query. The addition policies inject different types of non-existing triples. Lowest and Highest select candidates from the two extremes of the candidate-margin distribution, while Low-Mid, Mid, and Mid-High select candidates from progressively higher interior regions. Since KGs are incomplete, some non-existing candidates, especially those with stronger model support, may correspond to plausible

Table 2: Deletion: largest relative MRR **drops** from reference.

DB	Ratio	DistMult	ComplEx	Keci	MuRE	TransH	RotatE
Nations	0.02	-4.5%	-9.4%	-7.5%	-3.6%	-5.6%	-5.8%
	0.04	-7.1%	-9.5%	-7.4%	-6.2%	-7.2%	-3.2%
	0.06	-12.1%	-14.6%	-12.5%	-6.3%	-12.8%	-8.2%
	0.08	-10.1%	-17.7%	-15.0%	-9.4%	-12.9%	-6.7%
	0.10	-10.8%	-18.2%	-14.1%	-10.5%	-11.1%	-10.0%
KINSHIP	0.02	-3.5%	-5.3%	-6.1%	-6.2%	-2.4%	-4.0%
	0.04	-7.6%	-9.7%	-10.7%	-8.8%	-5.1%	-7.2%
	0.06	-8.6%	-12.8%	-11.5%	-13.0%	-13.0%	-9.7%
	0.08	-14.5%	-16.1%	-15.6%	-17.0%	-14.5%	-12.9%
	0.10	-12.8%	-19.6%	-17.9%	-15.8%	-14.6%	-13.3%
UMLS	0.02	-7.6%	-8.4%	-5.2%	-8.2%	-8.0%	-4.2%
	0.04	-12.5%	-9.0%	-11.9%	-12.8%	-11.5%	-6.7%
	0.06	-15.1%	-16.1%	-15.7%	-16.6%	-16.3%	-8.3%
	0.08	-19.0%	-17.7%	-19.8%	-20.4%	-20.0%	-14.1%
	0.10	-24.7%	-23.5%	-23.6%	-23.6%	-23.3%	-18.6%
FB15k-237	0.02	-4.6%	-5.6%	-10.1%	-6.5%	-1.3%	-1.8%
	0.04	-13.1%	-6.4%	-7.8%	-5.3%	-0.5%	-4.2%
	0.06	-14.9%	-10.2%	-23.7%	-11.9%	-8.2%	-6.5%
	0.08	-12.7%	-19.6%	-12.3%	-13.2%	-6.2%	-14.2%
	0.10	-17.1%	-11.2%	-60.9%	-14.0%	-7.1%	-18.0%
CoDEx	0.02	-0.9%	-9.5%	-9.4%	-7.7%	-4.2%	-5.9%
	0.04	-5.5%	-6.3%	-13.1%	-8.7%	-3.7%	-7.8%
	0.06	-10.2%	-9.8%	-17.2%	-12.1%	-5.7%	-8.8%
	0.08	-13.9%	-11.4%	-19.3%	-14.8%	-9.0%	-12.0%
	0.10	-15.3%	-18.1%	-19.1%	-17.7%	-16.2%	-17.0%
NELL-995 h100	0.02	-7.0%	-2.7%	-9.7%	-0.9%	-3.8%	-6.9%
	0.04	-7.0%	-8.9%	-20.0%	0.0%	-10.4%	-6.8%
	0.06	-7.0%	-11.2%	-14.1%	-3.6%	-12.0%	-8.7%
	0.08	-11.9%	-18.1%	-19.4%	-3.1%	-10.4%	-11.0%
	0.10	-12.9%	-12.0%	-11.5%	-6.5%	-11.6%	-13.7%
WN18RR	0.02	-13.4%	-1.6%	-36.6%	-5.4%	-9.6%	-7.4%
	0.04	-8.6%	-3.1%	-26.2%	-10.6%	-12.4%	-10.1%
	0.06	-11.4%	-14.4%	-33.3%	-13.6%	-12.9%	-14.4%
	0.08	-16.0%	-26.4%	-32.8%	-17.2%	-12.3%	-13.0%
	0.10	-17.9%	-22.3%	-37.3%	-21.5%	-14.7%	-18.1%

Random Lowest MidLow Mid MidHigh Highest

Table 3: Addition: largest relative MRR **drops** from reference.

DB	Ratio	DistMult	ComplEx	Keci	MuRE	TransH	RotatE
Nations	0.02	-4.0%	-6.4%	-7.3%	0.0%	-5.5%	-2.0%
	0.04	-3.1%	-9.8%	-4.9%	-4.7%	-3.7%	-0.8%
	0.06	-1.2%	-11.6%	-6.9%	-4.1%	-5.1%	-7.4%
	0.08	-3.4%	-14.5%	-6.2%	-6.4%	-3.3%	-7.0%
	0.10	-4.6%	-12.1%	-3.7%	-4.0%	-6.2%	-7.1%
KINSHIP	0.02	-1.7%	-2.8%	-4.4%	-1.5%	-2.1%	-0.2%
	0.04	-3.5%	-4.2%	-6.3%	-2.3%	-1.9%	-0.6%
	0.06	-4.3%	-5.1%	-6.4%	-3.5%	-4.3%	0.0%
	0.08	-6.1%	-6.3%	-7.9%	-6.6%	-7.4%	-0.2%
	0.10	-7.2%	-7.0%	-6.5%	-8.1%	-5.8%	-0.1%
UMLS	0.02	-3.4%	-1.3%	-2.7%	-0.1%	-3.5%	-3.0%
	0.04	-5.3%	-2.3%	-4.8%	-0.8%	-1.9%	-2.7%
	0.06	-4.0%	-4.0%	-8.4%	-2.1%	-4.4%	-6.7%
	0.08	-6.5%	-4.6%	-6.7%	-2.6%	-5.0%	-6.2%
	0.10	-7.3%	-4.2%	-7.3%	-2.5%	-5.7%	-5.3%
FB15k-237	0.02	-0.5%	-1.2%	-3.9%	-3.2%	-1.8%	0.0%
	0.04	-5.8%	-3.5%	-20.2%	-3.7%	-4.6%	-0.0%
	0.06	-3.6%	-11.6%	-2.7%	-4.2%	-2.9%	-1.7%
	0.08	-4.9%	-5.5%	-17.7%	-4.7%	-2.8%	-2.6%
	0.10	-6.3%	-6.0%	-9.4%	-5.7%	-3.5%	-1.7%
CoDEx	0.02	-1.7%	-3.2%	-4.9%	-3.9%	0.0%	-3.1%
	0.04	-2.5%	-3.0%	-7.6%	-4.0%	-3.0%	-5.6%
	0.06	-1.3%	-4.6%	-6.6%	-5.0%	-2.5%	-5.2%
	0.08	0.0%	-3.7%	-6.2%	-4.8%	-3.5%	-5.1%
	0.10	-0.1%	-6.3%	-7.2%	-5.7%	-3.5%	-6.8%
NELL-995 h100	0.02	-2.9%	-2.9%	-6.5%	0.0%	-5.0%	-0.1%
	0.04	-3.7%	-3.0%	-6.3%	-3.1%	-5.7%	-1.0%
	0.06	-7.4%	-4.7%	-8.5%	-3.0%	-10.7%	-2.4%
	0.08	-9.2%	-5.0%	-12.4%	-4.1%	-7.0%	-3.5%
	0.10	-10.3%	-6.0%	-12.1%	-5.3%	-10.7%	-3.1%
WN18RR	0.02	0.0%	-3.9%	-8.0%	-3.3%	-3.1%	-2.1%
	0.04	-5.0%	-2.6%	-6.6%	-5.3%	-7.6%	-3.5%
	0.06	-1.7%	-8.4%	-10.4%	-7.5%	-8.1%	-3.3%
	0.08	-3.6%	0.0%	-9.2%	-9.2%	-12.6%	-4.1%
	0.10	-3.6%	-1.5%	-9.0%	-12.1%	-14.2%	-4.9%

Rnd Lowest MidLow Mid MidHigh Highest

missing links rather than true negatives. At the same time, added triples may also distort the supervision signal if they introduce implausible facts into the training graph. Therefore, the effect of addition is not known in advance. Comparing random additions with separation-informed additions allows us to test whether the separation margin provides a useful signal for training-graph augmentation. If additions from particular empirical margin regions improve performance more often than random additions, the improvement cannot be attributed merely to increasing the number of training triples. Rather, it suggests that the choice of candidate additions matters, and that separation-margin analysis can distinguish candidate triples with different training effects. We examine these addition effects by separately analyzing cases where performance drops and cases where performance improves.

5.2.1 Damage Caused by Additions: The addition-damage shown in Table 3 provides a complementary view: some margin regions are associated with stronger performance degradation. The largest MRR drops are most often caused by Lowest and other low-margin additions. These candidates are generated for the same (h, r) prediction queries as the original triples, but they are weakly supported relative to competing tails. Treating them as positive training facts therefore injects supervision that conflicts with the model's ranking scheme and can distort relation-specific predictions. In contrast, Highest-region additions rarely produce the strongest damage, since they are more compatible with the learned ranking signal and may correspond to plausible missing links in an incomplete KG. Random additions are also rarely the most damaging. Thus, the damage is not caused merely by adding triples; it depends on which empirical margin region the added candidates come from.

5.2.2 Gains by Additions: Table 4 shows that the largest MRR improvements are most often obtained by adding triples from the Highest candidate-margin region. These additions are not arbitrary: for the same (h, r) query, they correspond to non-existing tails that the model ranks above most competing alternatives. Because knowledge graphs are incomplete, such candidates may represent plausible missing links and can provide useful additional supervision during training. Random addition uses the same budget but ignores this margin structure, and therefore rarely yields comparable gains. The observed improvements are thus not a consequence of adding more triples alone; they depend on selecting candidates from particular regions of the model's ranking landscape. This supports viewing the method as separation-guided self-training rather than generic graph augmentation.

6 Ensemble

The ensemble-based results show that the separation-margin viewpoint remains useful even when the target model's scores are not available. In this setting, perturbations are selected using proxy models, yet the deletion and addition patterns remain broadly consistent with the single-model results. This suggests that the signal captured by separation margins is not merely a property of one particular scoring function. Rather, it reflects a common structure of link-prediction systems: triples are assigned plausibility scores,

Table 4: Addition: largest relative MRR **gains** over reference.

DB	Ratio	DistMult	ComplEx	Keci	MuRE	TransH	RotatE
Nations	0.02	□ +0.7%	■ +1.5%	■ +4.0%	◆ +3.2%	■ +1.3%	■ +1.0%
	0.04	■ +0.3%	○ 0.0%	□ +1.5%	■ +3.6%	■ +5.1%	◆ +2.1%
	0.06	■ +2.1%	■ +1.7%	○ 0.0%	■ +4.6%	■ +10.1%	■ +1.5%
	0.08	■ +3.4%	■ +2.7%	■ +2.5%	■ +2.6%	■ +11.1%	■ +3.0%
	0.10	■ +4.3%	■ +1.0%	■ +4.6%	■ +4.0%	■ +18.2%	■ +2.2%
KINSHIP	0.02	■ +2.1%	■ +1.5%	○ 0.0%	■ +0.2%	■ +2.5%	■ +0.6%
	0.04	■ +3.7%	■ +1.6%	■ +1.0%	■ +0.6%	■ +5.9%	■ +1.6%
	0.06	■ +1.4%	■ +1.1%	■ +1.1%	■ +1.2%	■ +7.7%	■ +1.9%
	0.08	■ +2.0%	■ +1.7%	■ +0.5%	■ +1.7%	■ +5.4%	■ +2.1%
	0.10	■ +1.7%	■ +3.0%	■ +2.8%	■ +2.1%	■ +6.0%	■ +2.5%
UMLS	0.02	■ +1.0%	■ +2.1%	■ +2.8%	■ +1.5%	■ +1.6%	■ +0.0%
	0.04	■ +0.7%	■ +2.1%	■ +3.0%	■ +2.0%	■ +2.7%	■ +2.3%
	0.06	■ +0.7%	■ +3.1%	■ +4.3%	■ +1.6%	■ +2.7%	◆ +3.8%
	0.08	■ +1.2%	■ +4.0%	■ +3.8%	■ +1.9%	■ +3.4%	◆ +1.9%
	0.10	■ +0.2%	■ +4.4%	■ +4.7%	■ +2.2%	■ +3.6%	■ +2.8%
FB15k-237	0.02	■ +1.1%	■ +4.6%	◆ +5.1%	○ 0.0%	○ 0.0%	◆ +2.3%
	0.04	■ +3.0%	■ +4.1%	■ +2.2%	○ 0.0%	■ +0.7%	■ +0.9%
	0.06	■ +1.8%	■ +3.4%	◆ +3.5%	■ +0.4%	■ +1.5%	■ +0.9%
	0.08	■ +2.5%	■ +7.4%	◆ +2.4%	■ +3.5%	■ +0.6%	■ +1.3%
	0.10	■ +0.5%	◆ +2.7%	■ +5.9%	■ +0.9%	○ 0.0%	◆ +1.5%
CoDEx	0.02	■ +3.0%	■ +3.3%	■ +1.3%	○ 0.0%	■ +1.3%	■ +0.1%
	0.04	■ +5.1%	■ +5.4%	○ 0.0%	■ +2.0%	■ +3.1%	○ 0.0%
	0.06	■ +6.6%	■ +5.5%	■ +1.3%	■ +3.6%	■ +3.4%	○ 0.0%
	0.08	■ +7.0%	■ +8.8%	■ +2.9%	■ +4.3%	■ +1.8%	○ 0.0%
	0.10	■ +11.4%	■ +9.7%	■ +1.9%	■ +5.7%	■ +3.5%	■ +0.5%
NELL-995 h100	0.02	■ +1.7%	■ +2.9%	□ +0.1%	□ +2.6%	■ +1.6%	◆ +1.2%
	0.04	■ +1.0%	■ +5.9%	■ +0.4%	■ +5.4%	■ +1.6%	◆ +1.4%
	0.06	■ +3.4%	■ +6.1%	■ +1.5%	■ +3.8%	○ 0.0%	◆ +1.5%
	0.08	■ +2.4%	■ +8.9%	○ 0.0%	■ +7.6%	○ 0.0%	■ +1.1%
	0.10	■ +3.7%	■ +8.3%	■ +3.1%	■ +6.9%	■ +2.6%	○ 0.0%
WN18RR	0.02	◆ +3.4%	■ +15.1%	○ 0.0%	○ 0.0%	□ +0.8%	□ +0.2%
	0.04	◆ +2.1%	■ +6.2%	○ 0.0%	○ 0.0%	○ 0.0%	○ 0.0%
	0.06	■ +2.6%	◆ +11.1%	○ 0.0%	○ 0.0%	○ 0.0%	○ 0.0%
	0.08	○ 0.0%	■ +7.9%	○ 0.0%	○ 0.0%	○ 0.0%	○ 0.0%
	0.10	■ +0.2%	■ +7.5%	○ 0.0%	○ 0.0%	○ 0.0%	○ 0.0%

◆Rnd □ Lowest ■ MidLow ■ Mid ■ MidHigh ■ Highest

Table 5: Ensemble-based deletion: largest relative MRR **drops**.

DB	Ratio	DistMult	ComplEx	Keci	MuRE	TransH	RotatE
Nations	0.02	■ -3.9%	□ -8.8%	◆ -4.5%	■ -1.4%	■ -4.4%	◆ -5.4%
	0.04	□ -9.2%	□ -10.3%	■ -12.7%	■ -7.6%	■ -4.9%	□ -5.7%
	0.06	□ -11.4%	◆ -14.6%	■ -10.3%	■ -6.0%	■ -8.0%	■ -6.5%
	0.08	□ -13.6%	■ -15.8%	□ -14.7%	□ -9.3%	◆ -12.9%	□ -8.8%
	0.10	■ -16.3%	□ -16.4%	◆ -14.1%	■ -14.7%	■ -9.4%	■ -10.5%
KINSHIP	0.02	■ -4.6%	■ -5.3%	■ -5.8%	■ -5.7%	■ -4.3%	■ -4.1%
	0.04	■ -6.5%	■ -10.6%	◆ -10.1%	■ -7.9%	■ -4.8%	■ -5.6%
	0.06	■ -8.7%	■ -11.3%	■ -11.8%	■ -11.4%	■ -7.0%	■ -10.0%
	0.08	■ -10.6%	■ -16.5%	■ -14.7%	■ -12.6%	◆ -5.6%	■ -11.3%
	0.10	■ -15.2%	■ -18.7%	■ -19.3%	■ -17.7%	■ -10.3%	■ -10.4%
UMLS	0.02	■ -7.5%	■ -5.0%	■ -4.9%	■ -6.4%	■ -5.5%	■ -5.3%
	0.04	■ -12.1%	■ -11.4%	■ -9.2%	■ -11.5%	■ -12.5%	■ -8.6%
	0.06	■ -17.4%	■ -18.4%	■ -15.1%	■ -15.3%	■ -16.6%	■ -13.4%
	0.08	■ -19.6%	■ -19.0%	■ -17.9%	■ -19.5%	■ -21.0%	■ -17.2%
	0.10	◆ -24.7%	■ -21.8%	◆ -23.6%	■ -22.4%	■ -22.9%	■ -19.0%
FB15k-237	0.02	■ -6.3%	◆ -5.6%	◆ -10.1%	◆ -3.6%	■ -1.5%	■ -4.2%
	0.04	■ -6.8%	■ -5.6%	■ -43.6%	■ -5.2%	■ -4.2%	■ -4.6%
	0.06	■ -11.2%	■ -5.5%	■ -22.6%	■ -9.7%	■ -6.2%	■ -7.0%
	0.08	■ -13.1%	■ -14.1%	■ -37.6%	■ -9.5%	■ -8.6%	■ -9.9%
	0.10	■ -22.4%	■ -17.6%	◆ -60.9%	■ -9.9%	■ -11.9%	■ -13.1%
CoDEx	0.02	□ -5.8%	■ -5.9%	■ -11.2%	■ -5.3%	■ -4.3%	■ -4.0%
	0.04	■ -6.9%	■ -5.7%	◆ -13.1%	■ -8.6%	■ -7.0%	■ -7.1%
	0.06	◆ -7.9%	◆ -8.3%	■ -14.7%	■ -10.0%	■ -4.4%	■ -8.4%
	0.08	■ -12.2%	◆ -11.4%	■ -19.8%	■ -17.2%	■ -8.0%	■ -11.3%
	0.10	■ -10.8%	◆ -14.6%	◆ -19.1%	■ -15.5%	■ -8.5%	■ -12.9%
NELL-995 h100	0.02	◆ -4.6%	□ -7.8%	■ -8.7%	○ 0.0%	■ -2.5%	■ -6.2%
	0.04	■ -4.5%	◆ -8.9%	□ -14.1%	○ 0.0%	■ -6.8%	◆ -6.8%
	0.06	■ -5.2%	■ -12.2%	■ -13.0%	◆ -0.2%	■ -12.6%	■ -8.3%
	0.08	◆ -11.9%	◆ -18.1%	◆ -19.4%	■ -4.7%	■ -23.5%	■ -9.5%
	0.10	■ -12.3%	■ -16.0%	■ -25.3%	◆ -5.9%	■ -17.8%	■ -13.8%
WN18RR	0.02	□ -7.5%	◆ -16.1%	■ -22.5%	□ -11.3%	■ -6.2%	□ -4.8%
	0.04	□ -13.9%	■ -17.6%	◆ -35.2%	□ -15.9%	■ -9.6%	◆ -7.5%
	0.06	□ -19.7%	■ -27.9%	□ -28.1%	□ -19.1%	■ -12.8%	□ -11.8%
	0.08	□ -25.8%	□ -24.0%	□ -34.2%	□ -22.6%	■ -18.3%	□ -16.1%
	0.10	□ -30.6%	□ -32.8%	■ -33.2%	□ -27.3%	□ -17.0%	□ -17.4%

◆Rnd □ Lowest ■ MidLow ■ Mid ■ MidHigh ■ Highest

and their effect on training depends on how clearly they are distinguished from competing candidates. From this perspective, the ensemble approach does not need to reproduce the exact scores of the target link prediction system. It only needs to identify triples that occupy similar separation regions across link-prediction models, such as weakly separated, intermediate, and strongly separated triples. This suggest that separation-margin-guided perturbations reflect the ranking structure of knowledge graph completion, rather than only the behavior of individual models. The corresponding ensemble results are reported in Tables 5, 6, and 7.

7 Discussion

The separation-margin regions can be interpreted as different levels of alignment between asserted facts and the model's learned ranking of those facts relative to competing negative candidates. KGE models do not necessarily represent all observed training facts equally well. Many triples contribute to the same entity and relation embeddings. As a result, differences in graph connectivity, relation patterns, and model inductive bias may cause some facts to be represented more strongly than others. Our results suggest that removing weakly separated triples often has limited impact since these facts may already be only weakly reflected in the model's ranking. Strongly separated triples are also relatively robust to removal. This may be because strongly separated triples are supported by other training facts involving the same entities and relations. Intermediate-separation triples appear to be more influential. They are represented strongly enough to affect the ranking but are less robust to removal. We divide candidates into ordered regions, from weakly separated to strongly separated candidates. These regions reflect the model's degree of support, not the factual validity of the triples. A weakly separated candidate is therefore not necessarily false; it only means that the trained model does not represent this triple as clearly supported relative to competing tails for the same (h, r) query. In this sense, the fact is valid in the symbolic graph, but the trained model has not learned it strongly enough to separate it reliably from competing negative candidates. Such disagreement may arise because the chosen scoring function or embedding space is poorly suited to the relational pattern expressed by the fact. This interpretation also depends on the model family, since KGE models encode relational patterns differently [27]. Low separation may therefore reflect a limited representation of an underlying relational pattern rather than factual invalidity.

7.1 Interpreting Separation Margins

The effect of separation-guided perturbations may not be determined solely by the selected margin region. It can also depend on the structural properties of the dataset and on how the model scores competing candidate tails. A low separation margin, for example, does not necessarily mean that a triple is unimportant; it may instead indicate an ambiguous (h, r) query where several tails are plausible or where positives remain close to hard competing negatives. Thus, separation margins should be interpreted relative to the corresponding (h, r). We analyze how these signals vary across model families, tail cardinality, and perturbation outcomes.

Table 6: Ensemble-based addition: largest relative MRR **drops**.

DB	Ratio	DistMult	ComplEx	Keci	MuRE	TransH	RotatE
Nations	0.02	◆-3.2%	□-7.1%	□-6.0%	■-0.4%	□-4.2%	□-4.3%
	0.04	◆-3.1%	□-9.6%	◆-4.5%	□-1.4%	□-4.8%	□-8.2%
	0.06	■-4.2%	□-11.3%	■-8.6%	□-2.6%	■-5.2%	□-14.4%
	0.08	□-5.9%	□-11.8%	■-6.4%	□-4.6%	□-4.3%	□-11.2%
	0.10	■-6.1%	□-13.2%	□-7.7%	■-2.6%	■-3.5%	□-7.9%
KINSHIP	0.02	◆-1.2%	□-2.0%	□-2.9%	◆-1.5%	□-1.6%	■-1.1%
	0.04	□-1.8%	□-4.4%	■-4.2%	□-2.8%	□-0.9%	◆-0.6%
	0.06	□-4.7%	◆-4.8%	■-4.7%	□-4.1%	□-1.2%	■-0.0%
	0.08	□-6.1%	■-5.2%	□-6.3%	□-5.2%	□-0.3%	■-1.2%
	0.10	□-7.8%	■-6.0%	□-6.7%	□-5.3%	□-0.8%	○ 0.0%
UMLS	0.02	◆-2.5%	□-1.7%	■-2.9%	■-0.4%	■-2.6%	◆-3.0%
	0.04	□-4.5%	■-2.7%	■-3.6%	■-0.2%	■-3.8%	■-3.0%
	0.06	■-4.6%	◆-3.6%	◆-4.4%	■-0.4%	□-2.2%	□-0.8%
	0.08	■-6.3%	■-4.6%	■-4.6%	■-0.5%	□-2.2%	□-2.4%
	0.10	■-7.9%	□-6.4%	■-7.7%	◆-0.7%	□-2.2%	□-0.4%
FB15k-237	0.02	□-3.4%	■-4.3%	■-23.2%	■-2.4%	■-4.9%	■-0.1%
	0.04	□-4.3%	■-5.5%	□-16.6%	□-4.0%	□-4.3%	□-0.9%
	0.06	□-2.4%	■-4.2%	□-29.6%	◆-3.5%	□-5.3%	□-0.2%
	0.08	□-6.3%	□-1.1%	□-18.6%	■-3.5%	□-5.6%	□-1.2%
	0.10	□-9.8%	□-2.0%	■-13.1%	■-4.8%	□-7.7%	□-2.0%
CoDEx	0.02	○ 0.0%	○ 0.0%	■-7.2%	◆-3.4%	○ 0.0%	□-3.7%
	0.04	■-1.5%	□-2.2%	□-9.5%	□-5.2%	□-0.2%	◆-5.6%
	0.06	◆-1.3%	■-0.6%	□-9.6%	◆-4.4%	□-3.6%	□-5.7%
	0.08	□-2.8%	□-2.8%	□-9.4%	□-6.5%	□-4.6%	◆-4.6%
	0.10	□-2.0%	□-3.7%	□-8.6%	□-4.1%	□-3.3%	◆-6.6%
NELL-h100	0.02	□-2.8%	■-1.6%	■-8.9%	○ 0.0%	■-5.9%	○ 0.0%
	0.04	■-4.2%	◆-3.0%	■-9.5%	□-4.4%	□-4.5%	□-1.4%
	0.06	□-5.7%	◆-4.7%	◆-8.3%	□-4.3%	◆-10.7%	■-0.8%
	0.08	■-7.8%	□-5.9%	□-11.3%	□-2.0%	□-5.5%	■-0.2%
	0.10	□-8.7%	□-6.3%	◆-12.1%	■-4.6%	◆-10.7%	■-0.9%
WN18RR	0.02	■-1.6%	■-1.6%	□-2.8%	□-3.4%	◆-3.1%	□-2.1%
	0.04	○ 0.0%	◆-2.6%	■-10.9%	□-6.5%	□-5.1%	□-2.8%
	0.06	■-1.6%	□-6.8%	□-7.0%	■-7.9%	◆-6.4%	■-3.6%
	0.08	◆-0.5%	□-5.5%	□-9.3%	■-10.9%	◆-12.6%	■-3.8%
	0.10	◆-3.4%	■-6.1%	□-12.2%	■-12.5%	◆-12.2%	□-4.0%

◆Rnd □Lowest □MidLow ■Mid ■MidHigh ■Highest

Table 7: Ensemble-based addition: largest relative MRR **gains**.

DB	Ratio	DistMult	ComplEx	Keci	MuRE	TransH	RotatE
Nations	0.02	■+0.5%	■+3.2%	□+1.3%	◆+3.2%	□+2.5%	■+1.0%
	0.04	■+5.7%	■+1.7%	○ 0.0%	■+4.7%	◆+2.8%	■+3.2%
	0.06	■+2.2%	○ 0.0%	■+1.7%	■+4.1%	■+6.6%	■+1.4%
	0.08	■+1.2%	■+3.5%	■+3.9%	□+3.1%	■+3.6%	◆+2.8%
	0.10	■+4.4%	■+6.4%	■+9.4%	■+3.8%	■+3.9%	□+1.2%
KINSHIP	0.02	■+1.0%	○ 0.0%	■+0.4%	□+1.0%	■+1.6%	□+0.7%
	0.04	■+2.0%	○ 0.0%	○ 0.0%	■+0.8%	■+2.0%	■+0.9%
	0.06	■+3.2%	■+0.7%	■+2.7%	■+0.4%	■+1.4%	□+2.0%
	0.08	■+5.6%	○ 0.0%	■+1.6%	■+0.4%	■+2.0%	□+2.5%
	0.10	■+6.3%	■+2.0%	■+1.6%	■+2.5%	◆+2.3%	□+2.6%
UMLS	0.02	■+0.0%	■+0.2%	■+0.5%	■+0.6%	■+2.0%	■+1.2%
	0.04	■+0.2%	■+0.8%	■+2.1%	◆+1.0%	□+0.1%	□+3.3%
	0.06	■+1.6%	■+1.4%	■+2.2%	■+1.0%	◆+0.7%	◆+3.8%
	0.08	■+2.6%	■+2.2%	■+2.6%	■+1.2%	■+1.2%	◆+1.9%
	0.10	■+2.6%	■+3.4%	■+3.5%	■+1.0%	■+1.0%	■+3.3%
FB15k-237	0.02	□+1.7%	□+2.8%	◆+5.1%	■+0.7%	□+0.9%	◆+2.3%
	0.04	■+0.7%	◆+2.5%	■+3.5%	■+0.7%	◆+0.5%	■+1.9%
	0.06	■+3.0%	■+3.6%	◆+3.5%	■+0.2%	■+1.6%	■+2.1%
	0.08	■+7.9%	■+9.5%	□+5.5%	■+3.0%	○ 0.0%	■+2.3%
	0.10	■+3.6%	■+9.1%	■+7.9%	■+3.6%	■+0.1%	■+1.5%
CoDEx	0.02	□+2.9%	◆+2.9%	○ 0.0%	■+0.9%	■+2.1%	■+1.2%
	0.04	■+5.2%	□+5.0%	○ 0.0%	■+1.9%	■+2.6%	■+0.7%
	0.06	■+5.5%	■+5.8%	○ 0.0%	○ 0.0%	□+0.5%	○ 0.0%
	0.08	■+7.9%	■+7.1%	○ 0.0%	■+1.2%	◆+0.8%	■+2.5%
	0.10	■+9.8%	■+10.4%	■+1.4%	■+1.6%	■+0.4%	■+2.6%
NELL-995 h100	0.02	■+1.5%	■+4.1%	■+2.9%	■+6.6%	○ 0.0%	■+1.2%
	0.04	■+6.8%	■+10.2%	■+5.7%	■+4.6%	○ 0.0%	■+1.5%
	0.06	■+6.7%	■+12.4%	■+11.9%	■+7.4%	○ 0.0%	■+2.7%
	0.08	■+10.2%	■+14.5%	■+14.4%	■+11.9%	■+2.9%	□+0.6%
	0.10	■+12.9%	■+20.0%	■+17.0%	■+12.2%	■+1.0%	□+1.6%
WN18RR	0.02	■+8.1%	■+21.3%	■+4.0%	○ 0.0%	■+5.2%	■+3.7%
	0.04	■+10.3%	■+24.7%	■+5.2%	■+1.3%	■+8.0%	■+6.4%
	0.06	■+9.0%	■+19.4%	■+7.4%	■+0.5%	■+8.7%	■+7.2%
	0.08	■+9.0%	■+18.5%	■+6.5%	■+1.3%	■+7.2%	■+7.5%
	0.10	■+9.0%	□+8.7%	■+8.4%	■+0.2%	■+4.9%	■+6.6%

◆Rnd □Lowest □MidLow ■Mid ■MidHigh ■Highest

7.1.1 Model's Effect: Figure 2 shows that the separation-margin distributions vary across models. DistMult [33], ComplEx [29], Keci [9], and MuRE [3] often produce broader margin distributions, especially on sparse datasets such as CoDEx-S [24], NELL-995-h100 [32], and WN18RR [11]. In contrast, TransH [31] and RotatE [27] produce narrower distributions that remain closer to the hard-negative boundary. These differences indicate that the scale and distribution of separation margins are model dependent. TransH [31] scores triples after projecting entities onto relation-specific hyperplanes, whereas RotatE [27] represents relations as rotations in complex space. Their different scoring functions therefore induce different score distributions and notions of distance between positive and competing tails. We consequently interpret the narrower margins of TransH [31] and RotatE [27] as model-specific characteristics of the learned scores rather than as evidence of weaker learning. These model-specific differences also motivate our ensemble formulation. Rather than directly combining raw separation margins, which may not be comparable across models, we first convert the margins produced by each model into within-model percentile ranks. The ensemble then aggregates these ranks to capture the consensus relative position of each triple across models. Thus, although the absolute scale and distribution of separation margins differ across models, there is sufficient agreement in the relative positioning of triples for a consensus ranking to be useful.

7.1.2 Dataset's Effect: Figure 4 further shows that median separation margins vary with tail cardinality. This quantity denotes the number of training tails associated with the same (h, r) query. Higher tail cardinality means that the model must rank several valid or plausible tails in the same prediction context. In incomplete KGs, some plausible tails may be absent from the observed positives and therefore appear among the candidate negatives. If these tails receive high scores, they can shift the upper part of the negative-score distribution upward. This can increase the hard-negative boundary and consequently reduce the separation margin. This is particularly important for WN18RR [11]. As shown in Figure 3, WN18RR [11] is not only sparse; it is also dominated by 1-to-N/N-to-1 relation patterns. Such patterns create local prediction contexts with strong competing alternatives. Therefore, a low or intermediate separation margin in WN18RR [11] does not necessarily mean that a triple is unimportant. It indicates that the triple is associated with a relation for which plausible alternatives are difficult to separate.

7.1.3 Implications for deletion and addition: The observations in Sections 7.1.1 and 7.1.2 help explain why deletion and addition effects differ across models and datasets. In models with broader separation distributions, empirical margin regions are more clearly separated. Intermediate-margin triples can be especially influential because they are learned well enough to affect ranking decisions, yet remain close enough to competing alternatives that removing them can alter the local ranking structure. For TransH [31] and RotatE [27] , the same regions are compressed in raw margin space. Deletion effects can therefore differ because the model keeps positives and hard negatives closer together. The same reasoning applies to addition. High-margin additions often improve performance because they are selected from candidates that the model ranks highly

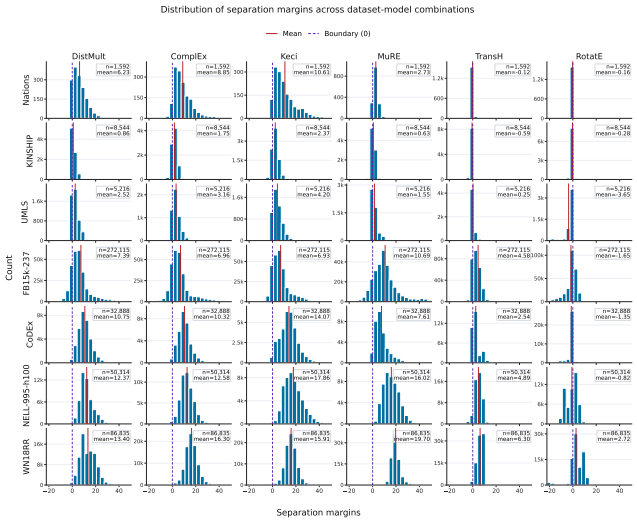


Figure 2: Separation-margin distributions across dataset–model pairs; red: mean, dashed: zero boundary.

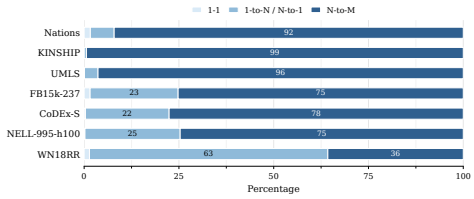


Figure 3: Percentage of training triples associated with each relation cardinality type across datasets.

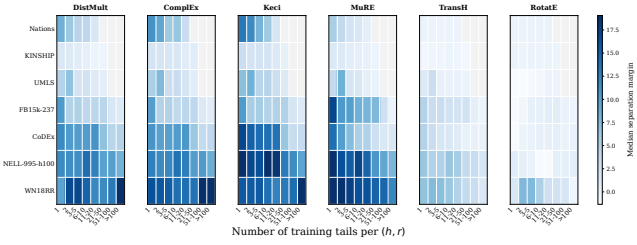


Figure 4: Median separation margins by tail cardinality across datasets and models. Margins vary with cardinality structure and KGE model.

within the same (h, r) context, and some of them can correspond to plausible missing links. However, for TransH [31] and RotatE [27], high-margin candidates occur within a more compressed score landscape. Such candidates can help when they are missing facts, but they can also disturb the learned ranking structure when they are plausible but false alternatives. Tail cardinality explains part of the variation across datasets, while the model-specific patterns observed for TransH [31] and RotatE [27] indicate that the hard-negative boundary used in the separation-margin calculation is influenced by each model's scoring characteristics.

8 Conclusion

We introduced separation-margin-guided additions and deletions as controlled probes to assess how training triples influence KGE link-prediction rankings. The separation margin reveals whether the model supports a fact strongly, weakly, or moderately relative to competing alternatives. Our results show that link-prediction rankings depend unevenly on training triples. For deletion, triples from specific separation-margin bands are more influential than random or extreme-margin removals. Some triples therefore appear to contribute more strongly to maintaining the ranking of triples. Adding weakly supported candidates often harms performance while higher-margin candidates can improve it. This suggests that augmentation depends on compatibility with the model's ranking and KG structure. Ensemble results support this interpretation. This indicates that separation-margin-guided edits capture recurring ranking vulnerabilities beyond a single trained model. In summary, separation-margin analysis provides a diagnostic tool for understanding and improving KGE robustness. It highlights influential training facts and candidate additions that can harm or improve performance. Perturbation therefore becomes a tool for probing and improving link prediction. Finally, our perturbation datasets provide a benchmark for poisoning, robustness, and augmentation in KG completion.

9 Future Work

Future work can extend separation-margin perturbations to mixed edits that combine deletion and addition under a fixed budget. One option is replacement-style editing within the same (h, r) query by removing (h, r, t) from one margin region and adding (h, r, t') from another. Ensemble selection can also be extended through voting or consensus across multiple surrogate models and compared with single-surrogate selection.

10 Reproducibility

To ensure reproducibility, we made our data and code publicly available at https://doi.org/10.5281/zenodo.20363079. The archive contains the benchmark knowledge graph training datasets and all scripts needed to train reference KGE models, compute separation margins, generate addition and deletion perturbations, run direct and ensemble-based experiments, and evaluate the performance. It also includes a `README.txt` with installation instructions, directory structure, and commands for running the experiments.

11 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. We acknowledge the support of the German Federal Ministry of Research, Technology and Space (BMFTR) within the project KI-OWL under grant no. 01IS24057B.

12 GenAI Disclosure

To improve the clarity and readability of the paper, Gemini and Grammarly were occasionally used solely for language polishing. No scientific content, data analysis, results, interpretations, or conclusions were generated or developed using GenAI.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. Square Attack: A Query-Efficient Black-Box Adversarial Attack via Random Search. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII (Lecture Notes in Computer Science)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 484–501. doi:10.1007/978-3-030-58592-1_29

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007 (Lecture Notes in Computer Science, Vol. 4825)*. Springer, 722–735. doi:10.1007/978-3-540-76298-0_52

[3] Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. Multi-relational poincaré graph embeddings. *Advances in neural information processing systems* 32 (2019).

[4] Patrick Betz, Christian Meilicke, and Heiner Stuckenschmidt. 2022. Adversarial Explanations for Knowledge Graph Embeddings. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, Luc De Raedt (Ed.). ijcai.org, 2820–2826. doi:10.24963/IJCAI.2022/391

[5] Peru Bhardwaj, John D. Kelleher, Luca Costabello, and Declan O'Sullivan. 2021. Adversarial Attacks on Knowledge Graph Embeddings via Instance Attribution Methods. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 8225–8239. doi:10.18653/V1/2021.EMNLP-MAIN.648

[6] Peru Bhardwaj, John D. Kelleher, Luca Costabello, and Declan O'Sullivan. 2021. 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, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 1875–1888. doi:10.18653/v1/2021.acl-long.147

[7] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, Bhavani Thuraisingham, Battista Biggio, David Mandell Freeman, Brad Miller, and Arunesh Sinha (Eds.). ACM, 15–26. doi:10.1145/3128572.3140448

[8] Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). ACM, 365–374. doi:10.1145/2600428.2609628

[9] Caglar Demir and Axel-Cyrille Ngomo. 2023. Clifford Embeddings–A Generalized Approach for Embedding in Normed Algebras. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.

[10] Woodrow W Denham. 1973. *The detection of patterns in Alyawara nonverbal behavior*. Ph. D. Dissertation. University of Washington.

[11] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[12] David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. 2010. Building Watson: An Overview of the DeepQA Project. *AI Mag.* 31, 3 (2010), 59–79. doi:10.1609/AIMAG.V31I3.2303

[13] Emma J. Gerritse, Faegheh Hasibi, and Arjen P. de Vries. 2025. Graph Embeddings to Empower Entity Retrieval. *Inf. Retr. Res. J.* 1, 1 (2025), 137–165. doi:10.54195/IRRJ.19877

[14] Chuan Guo, Jacob R. Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q. Weinberger. 2019. Simple Black-box Adversarial Attacks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 2484–2493. http://proceedings.mlr.press/v97/guo19a.html

[15] Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. 2016. Vandalism Detection in Wikidata. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, Snehasis Mukhopadhyay, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, Yi Chang, Yunyao Li, and Parikshit Sondhi (Eds.). ACM, 327–336. doi:10.1145/2983323.2983740

[16] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box Adversarial Attacks with Limited Queries and Information. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 2142–2151. http://proceedings.mlr.press/v80/ilyas18a.html

[17] Sourabh Kapoor, Arnab Sharma, Michael Röder, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2025. Robustness Evaluation of Knowledge Graph Embedding Models Under Non-targeted Attacks. In *The Semantic Web - 22nd European Semantic Web Conference, ESWC 2025, Portoroz, Slovenia, June 1-5, 2025, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 15718)*, Edward Curry, Maribel Acosta, María Poveda-Villalón, Marieke van Erp, Adegboyega K. Ojo, Katja Hose, Cogan Shimizu, and Pasquale Lisena (Eds.). Springer, 264–281. doi:10.1007/978-3-031-94575-5_15

[18] Zhenfeng Lei. 2017. KGDatasets: Knowledge Graph Dataset Collection. https://github.com/ZhenfengLei/KGDatasets. Accessed: 2026-05-23.

[19] Alexa T McCray. 2003. An upper-level ontology for the biomedical domain. *Comparative and Functional genomics* 4, 1 (2003), 80–84.

[20] Adel Memariani, Michael Röder, Arnab Sharma, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2025. Link Prediction Under Non-targeted Attacks: Do Soft Labels Always Help?. In *The Semantic Web - ISWC 2025 - 24th International Semantic Web Conference, Nara, Japan, November 2-6, 2025, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 16140)*, Daniel Garijo, Sabrina Kirrane, Angelo A. Salatino, Cogan Shimizu, Maribel Acosta, Andrea Giovanni Nuzzolese, Sebastián Ferrada, Thibaut Soulard, Kouji Kozaki, Hideaki Takeda, and Anna Lisa Gentile (Eds.). Springer, 99–121. doi:10.1007/978-3-032-09527-5_6

[21] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[22] PyKEEN Developers. 2026. PyKEEN: A Python Library for Knowledge Graph Embeddings. https://github.com/pykeen/pykeen. Accessed: 2026-05-23.

[23] Hao Qiu, Leonardo Lucio Custode, and Giovanni Iacca. 2021. Black-box adversarial attacks using evolution strategies. In *GECCO '21: Genetic and Evolutionary Computation Conference, Companion Volume, Lille, France, July 10-14, 2021*, Krzysztof Krawiec (Ed.). ACM, 1827–1833. doi:10.1145/3449726.3463137

[24] Tara Safavi and Danai Koutra. 2020. CoDEx: A Comprehensive Knowledge Graph Completion Benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 8328–8350. doi:10.18653/V1/2020.EMNLP-MAIN.669

[25] Arnab Sharma, N'Dah Jean Kouagou, and Axel-Cyrille Ngonga Ngomo. 2025. Resilience in Knowledge Graph Embeddings. *Trans. Graph Data Knowl.* 3, 2 (2025), 1:1–1:38. doi:10.4230/TGDK.3.2.1

[26] Arnab Sharma, N'Dah Jean Kouagou, and Axel-Cyrille Ngonga Ngomo. 2025. Resilience in Knowledge Graph Embeddings. *Trans. Graph Data Knowl.* 3, 2 (2025), 1:1–1:38. doi:10.4230/TGDK.3.2.1

[27] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197* (2019).

[28] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*. 57–66.

[29] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*. PMLR, 2071–2080.

[30] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85. doi:10.1145/2629489

[31] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 28.

[32] Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. *arXiv preprint arXiv:1707.06690* (2017).

[33] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* (2014).

[34] Guangqian Yang, Lei Zhang, Yi Liu, Hongtao Xie, and Zhendong Mao. 2025. Exploiting Pre-Trained Language Models for Black-Box Attack against Knowledge Graph Embeddings. *ACM Trans. Knowl. Discov. Data* 19, 1 (2025), 1:1–1:14. doi:10.1145/3688850

[35] Xiaoyu You, Beina Sheng, Daizong Ding, Mi Zhang, Xudong Pan, Min Yang, and Fuli Feng. 2023. MaSS: Model-agnostic, Semantic and Stealthy Data Poisoning

Attack on Knowledge Graph Embedding. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 2000–2010. doi:10.1145/3543507.3583203

[36] Hengtong Zhang, Tianhang Zheng, Jing Gao, Chenglin Miao, Lu Su, Yaliang Li, and Kui Ren. 2019. Data Poisoning Attack against Knowledge Graph Embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 4853–4859. doi:10.24963/IJCAI.2019/674

[37] Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Chao Li, Hui Xiong, Qing He, and Yongjun Xu. 2023. Towards Robust Knowledge Graph Embedding via Multi-Task Reinforcement Learning. *IEEE Trans. Knowl. Data Eng.* 35, 4 (2023), 4321–4334. doi:10.1109/TKDE.2021.3127951

[38] Tianzhe Zhao, Jiaoyan Chen, Yanchi Ru, Qika Lin, Yuxia Geng, and Jun Liu. 2024. Untargeted Adversarial Attack on Knowledge Graph Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.). ACM, 1701–1711. doi:10.1145/3626772.3657702