

PERU BHARDWAJ

ADVERSARIAL ROBUSTNESS OF  
REPRESENTATION LEARNING FOR KNOWLEDGE GRAPHS





**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

School of Computer Science and Statistics

# Adversarial Robustness of Representation Learning for Knowledge Graphs

PERU BHARDWAJ  
bhardwap@tcd.ie

September 2022

A Thesis submitted in partial fulfilment  
of the requirements for the degree of  
Doctor of Philosophy

Peru Bhardwaj:

*Adversarial Robustness of Representation Learning for Knowledge Graphs,*

© September 2022

---

## DECLARATION

---

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

*Dublin, Ireland, September 2022*

---

Peru Bhardwaj



---

## ABSTRACT

---

Knowledge graphs represent factual knowledge about the world as relationships between concepts and are critical for intelligent decision making in enterprise applications. New knowledge is inferred from the existing facts in the knowledge graphs by encoding the concepts and relations into low-dimensional feature vector representations. The most effective representations for this task, called Knowledge Graph Embeddings (KGE), are *learned* through neural network architectures. Due to their impressive predictive performance, they are increasingly used in high-impact domains like healthcare, finance and education. However, are the black-box KGE models adversarially robust for use in domains with high stakes?

This thesis argues that state-of-the-art KGE models are vulnerable to data poisoning attacks, that is, their predictive performance can be degraded by systematically crafted perturbations to the training knowledge graph. To support this argument, two novel data poisoning attacks are proposed that craft input deletions or additions at training time to subvert the learned model's performance at inference time. These attacks target the task of predicting the missing facts in knowledge graphs using Knowledge Graph Embeddings.

To degrade the model performance through adversarial deletions, the use of model agnostic instance attribution methods is proposed. These methods are used to identify the training instances that are most influential to the KGE model's predictions on target instances. The influential triples are used as adversarial deletions. To poison the KGE models through adversarial additions, their inductive abilities are exploited. The inductive abilities of KGE models are captured through the relationship patterns like symmetry, inversion and composition in the knowledge graph. Specifically, to degrade the model's prediction confidence on target facts, this thesis proposes to improve the model's prediction confidence on a set of decoy facts. Thus, adversarial additions that can improve the model's prediction confidence on decoy facts through different relation inference patterns are crafted.

Evaluation of the proposed adversarial attacks shows that they outperform state-of-the-art baselines against four KGE models for two publicly available datasets. Among the proposed methods, simpler attacks are competitive with or outperform the computationally expensive ones. The thesis contributions not only highlight and provide an opportunity to fix the security vulnerabilities of KGE models, but also help to understand the black-box predictive behaviour of these models.





---

## ACKNOWLEDGEMENT

---

This research was conducted with the financial support of Accenture Labs and Science Foundation Ireland (SFI) at the ADAPT SFI Research Centre at Trinity College Dublin (TCD). The ADAPT SFI Centre for Digital Content Technology is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant No. 13/RC/2106 P2.

A fees-free extension for the submission of thesis was supported by the Office of the Dean of Graduate Studies at TCD, and research costs by the Irish Higher Education Authority (HEA) through Trinity HEA Covid Extension Fund 2020.

The research project was supervised by Prof. Declan O'Sullivan (ADAPT Centre, TCD); and co-supervised by Dr. Luca Costabello (Accenture Labs) and Prof. John Kelleher (ADAPT Centre, TU Dublin). The project was also co-supervised, at different stages by Dr. Fabrizio Orlandi (ADAPT Centre, TCD), and Dr. Freddy Lecue (Accenture Labs). The order is based on the length of involvement in the project.

Thanks to the project sponsors and supervisors for facilitating the research.  
Thank you to colleagues, housemates, friends and family for support and companionship over the years.



---

# CONTENTS

---

1	INTRODUCTION	1
1.1	Motivation	1
1.2	Research Problem	4
1.3	Research Contributions	6
1.4	Publications	8
1.5	Thesis Outline	9
2	REPRESENTATION LEARNING FOR KNOWLEDGE GRAPHS	10
2.1	Knowledge Graphs	11
2.2	Knowledge Graph Embeddings	13
2.3	Scoring Functions	13
2.4	Characteristics of Scoring Functions	17
2.4.1	Multiplicative vs Additive Interactions	17
2.4.2	Inductive Abilities of Knowledge Graph Embeddings	17
2.5	Training Knowledge Graph Embeddings	18
2.5.1	Training Strategies	19
2.5.2	Loss Functions	19
2.6	Missing Link Prediction	21
2.6.1	Evaluation Protocol	22
2.6.2	Evaluation Metrics	22
2.6.3	Benchmark Datasets	23
2.7	Summary	25
3	PROBLEM FORMULATION	26
3.1	Adversarial Attacks	27
3.2	Threat Model for Adversarial Attacks	28
3.2.1	Attacker's Goal	29
3.2.2	Attacker's Knowledge	30
3.2.3	Attacker's Capability	30
3.3	Design Space for Adversarial Attacks	31
3.3.1	Challenges	34
3.3.2	Design Requirements	35
3.4	State-of-the-art Solutions	36

## CONTENTS

3.4.1	Adversarial Deletions	37
3.4.2	Adversarial Additions	38
3.5	Proposed Attack Strategy	40
3.6	Evaluation Protocol	41
3.6.1	Evaluation Metrics	43
3.6.2	Baselines	43
3.6.3	Models and Datasets	45
3.7	Summary	45
4	POISONING VIA INSTANCE ATTRIBUTION METHODS	47
4.1	Threat Model	47
4.2	Instance Attribution Methods	48
4.3	Adversarial Additions	53
4.4	Experimental Setup	53
4.4.1	Datasets	54
4.4.2	Baselines	55
4.5	Evaluation of Attack Effectiveness	55
4.5.1	Comparison with Baselines	60
4.5.2	Comparison across Instance Attribution Metrics	61
4.5.3	Comparison of Datasets	62
4.6	Additional Evaluation	62
4.6.1	Analysis of Runtime Efficiency	62
4.6.2	Qualitative Analysis	64
4.7	Summary	68
5	POISONING VIA RELATION INFERENCE PATTERNS	71
5.1	Threat Model	71
5.2	Relation Inference Patterns	72
5.3	Steps for Adversarial Attack	73
5.3.1	Step1: Determine Adversarial Relations	75
5.3.2	Step2: Determine Decoy Entities	76
5.3.3	Step3: Determine Adversarial Entities	78
5.4	Computational Complexity Analysis	78
5.5	Experimental Setup	81
5.5.1	Datasets	81
5.5.2	Baselines	81
5.5.3	Implementation	82
5.6	Evaluation of Attack Effectiveness	83
5.6.1	Comparison across Relation Inference Patterns	85

5.6.2	Comparison of Datasets	85
5.6.3	Analysis on Decoy Triples	86
5.7	Additional Evaluation	87
5.7.1	Analysis of Runtime Efficiency	87
5.7.2	Qualitative Analysis	89
5.7.3	Comparison with Attribution Attacks	91
5.8	Summary	97
6	RELATED WORK	98
6.1	Graph Representation Learning	98
6.1.1	Based on Edge Reconstruction	98
6.1.2	Based on Neighbourhood Aggregation	100
6.1.3	Based on Random Walks	101
6.2	Understanding Knowledge Graph Representations	102
6.2.1	Via Robustness of Model Predictions	102
6.2.2	Via Explainability of Model Predictions	104
6.2.3	Via Miscellaneous Methods	105
6.3	Summary	106
7	CONCLUSION	107
7.1	Contributions	107
7.2	Limitations	108
7.2.1	Evaluation Design	109
7.2.2	Design of Adversarial Attacks	110
7.3	Broader Impact	110
7.4	Future Directions	112
7.4.1	Additional Evaluation	112
7.4.2	Additional Threat Models	112
7.4.3	Adversarially Robust Knowledge Graph Embeddings	113
7.5	Concluding Statement	113
A	ADDITIONAL DETAILS FOR INSTANCE ATTRIBUTION ATTACKS	115
A.1	Training Details	115
A.1.1	Training KGE models	115
A.1.2	Baseline Implementation Details	116
A.2	Additional Comparison with CRIAGE	118
B	ADDITIONAL DETAILS FOR INFERENCE PATTERN ATTACKS	119
B.1	Implementation Details	119

CONTENTS

B.1.1	Training KGE models	119
B.1.2	Baseline Implementation Details	120
B.1.3	Attack Implementation Details	121

BIBLIOGRAPHY	123
--------------	-----

---

## INTRODUCTION

---

### 1.1 MOTIVATION

Consider a financial regulator that wants to detect and anticipate money laundering activities, where malicious parties funnel their income from illicit activities into legitimate bank accounts through a series of financial transactions. Given the complex and interconnected nature of such activities, analysing the financial transactions from isolated bank accounts is often insufficient to catch fraudsters in practice. Rather, the relationships between different financial entities are leveraged to perform an integrated analysis across diverse sources of financial data. To enable this integrated analysis, knowledge graphs have emerged as the de-facto standard for modeling and integrating the factual knowledge about multiple financial entities from diverse sources (Hogan et al., 2021; Noy et al., 2019). Entities like bank accounts, their owners and their assets are represented as nodes of the knowledge graph, and the interactions and transactions between these entities as the labelled edges of the graph. Due to the growing volume of digital financial transactions, data-driven Machine Learning (ML) methods are used to mine the patterns in this interconnected graph of transactions. Learning and inference on the graph data drives intelligent decisions like predicting the suspicious transactions between different accounts or hidden affiliations between the account owners (Khalili et al., 2020; Chang, 2020; Singson and Soni, 2021). However, malicious parties with illicit income to launder would be highly motivated to escape detection from these systems. Thus, such parties might attempt to sabotage the predictions of data-driven intelligent systems by manipulating their personal details and transactions in the input knowledge graph. *Are the predictions from Machine Learning models on knowledge graphs reliable for use in such adversarial settings?*

Like the anti-money laundering application in finance, knowledge graphs are a ubiquitous representation of the factual knowledge about interconnected entities and the relationships between them (Hogan et al., 2021). In recent years, several

large-scale knowledge graphs have been developed to support intelligent decision making for applications ranging from search engines, e-commerce and social networks to biomedicine and finance. Commercial enterprises like Google and Microsoft have built web-scale knowledge graphs from textual sources on the Web to support Google Search and Bing. Similarly, Facebook and LinkedIn rely on the graph representation of the knowledge about their users to understand user preferences and recommend prospective connections or job opportunities. Enterprise knowledge graphs are also used by online vendors like Amazon and eBay to encode shopping behaviour of their users and information of their products for improved product recommendations. Other companies like Accenture, Deloitte and Bloomberg have deployed knowledge graphs for financial services. These financial graphs power applications like enterprise search, financial data analytics, risk assessment and fraud detection (Hogan et al., 2021; Noy et al., 2019).

More generally, in the domain of Natural Language Processing (NLP), background knowledge represented as factual knowledge graphs is injected to support knowledge-aware applications, like knowledge based question answering or explainable fact checking over knowledge bases (Ji et al., 2022; Kotonya and Toni, 2020). An emerging research direction in this domain investigates methods to combine the structured knowledge representation with the language representation learned from unstructured text. These methods seek to improve the contextual reasoning and understanding capabilities of NLP models using factual as well as commonsense knowledge (Malaviya et al., 2020; He et al., 2020; Zhang et al., 2022). Similarly, in the domain of Computer Vision (CV), machine learning for tasks like image classification, Visual Question Answering and skeleton based action recognition, is enhanced by representing the relationships among the objects in a scene or an image as knowledge graphs (see Ma and Tang, 2021, Chapter 11). On the other hand, there are emerging applications of knowledge graphs in healthcare and biomedical research. Here, biological networks as used to model the associations and interactions between different protein structures, drugs and diseases. Additionally, combining the biomedical knowledge with patients' electronic health records enables an integrated analysis of disease comorbidity for personalised precision medicine (Rotmensch et al., 2017; Mohamed et al., 2020; Li et al., 2021; Bonner et al., 2022). Thus, knowledge graphs are the backbone for learning and inference in modern intelligent systems.

To incorporate the graph data into the standard ML pipeline, the symbolic graph structure needs to be represented as differentiable feature vectors. Traditional algorithms used heuristics and domain engineering to hand-craft the features based on the statistics about graph or node topology, or the kernel methods (see Hamil-



ton, 2020, Chapter 2). The lack of flexibility of feature engineering has led way to the *graph representation learning* algorithms that *learn* to represent the graph structure as low-dimensional continuous feature vectors, also called embeddings. To learn the embedding for an entity, these algorithms aim to preserve the structural information about the entity's neighbourhood in the graph domain as a similarity measure in the embedding domain. This way, algebraic operations on the embeddings of entities reflect the graph structural interactions between those entities, which in turn, allows the topological information from the graph to be used for different ML tasks. Deep learning based neural network architectures are used to optimize the representations of entities and relations in the knowledge graph, such that the learned representations best support the downstream ML task to be performed on the graph (Hamilton et al., 2017b; Hamilton, 2020, Part 1). Due to their effectiveness for different downstream tasks, graph representation learning algorithms have become state-of-the-art methods for learning and reasoning with knowledge graphs (Chen et al., 2020; Nickel et al., 2016a).

The success of deep learning based methods is ascribed to their ability to extract rich statistical patterns from large volumes of input data. However, being data-driven, the learned models are non-interpretable and the reasons for their predictions are unknown. Due to this black-box predictive behaviour, the failure modes of the models are also unknown. It has been shown that the predictions from deep learning models can be manipulated by manipulating their input data (Biggio and Roli, 2018; Joseph et al., 2019). This is especially concerning for high-stakes domains like healthcare, finance, education or law enforcement where the representation learning algorithms for knowledge graphs are increasingly used (Mohamed et al., 2020; Bonner et al., 2022). In these domains, decision outcomes impact human lives and the stakes for model failure are very high. On the other hand, due to the high-stakes, there are likely going to be motivated adversarial actors that want to manipulate the model predictions. Additionally, knowledge graphs are often automatically extracted from textual sources on the Web or curated from user generated content (Nickel et al., 2016a; Ji et al., 2022). This makes it easy for the adversaries to inject carefully crafted false data to the graph.

Thus, to deploy graph representation learning models in high-stakes user facing domains, impressive predictive performance of the models is not enough. It is vital to ensure that the models are also safe and robust for use. Yet, building adversarially robust models requires methods to measure the adversarial robustness of a model. In other words, a necessary pre-requisite for adversarially robust graph representation learning models are methods to identify the failure modes or the security vulnerabilities of existing models.

*The research in this thesis is motivated by the need to identify the adversarial vulnerabilities of black-box graph representation learning algorithms as a crucial step towards their responsible integration in high-stakes user facing applications.*

## 1.2 RESEARCH PROBLEM

The research in Machine Learning (ML) aims to build intelligent computer applications that learn to perform real-world tasks from past experience instead of being explicitly programmed by human operators (Goodfellow et al., 2016). ML systems automatically recognize and extract patterns from the past evidence provided to the learner in the form of sample data. The patterns and assumptions observed in the input data are captured implicitly in a latent model that enables inferences on the unseen data. This predictive model design has been used successfully to make intelligent decisions from input data modalities like images, text and graphs. Because of the ability of the learner to adapt to input evidence, Machine Learning has emerged as the most promising approach to tackle real-world problems. With the availability of large volumes of training data, these systems now have applications in several daily tasks (Barreno et al., 2010; Biggio and Roli, 2018).

However, the adaptability of ML systems to input data can also become a vulnerability in real-world settings. In these settings, there might be motivated malicious actors that attempt to exploit the adaptive nature of the learning system to manipulate its predictions. The adversary can carefully design training data samples that steer the model to learn patterns that result in manipulated predictions. Such intentional perturbations of an ML model’s input data, that aim to manipulate the model predictions, constitute the security threat called *adversarial attacks*. The security analysis of ML systems is the focus of the research field of Adversarial Machine Learning (AML), that lies at the intersection of the wider fields of Machine Learning and Computer Security. The latter advocates for a proactive approach to system security, also called security by design. Instead of designing countermeasures in response to the adversarial attacks on deployed systems, the proactive approach anticipates these attacks during system design and evaluates the system’s resilience (that is, its adversarial robustness) to the simulated attacks (Biggio and Roli, 2018; Joseph et al., 2019; Barreno et al., 2010).

This design approach has been widely used to investigate the adversarial robustness of ML systems for domains like images and natural language. However, adversarial attacks designed against ML systems for images or text are not directly applicable to graph data. This is because of the discrete perturbation space for the graph domain and the interconnected nature of the graph structure (Günnemann,

2022). There have been recent efforts towards investigating the adversarial robustness of machine learning for graph data, as surveyed in [Günemann \(2022\)](#), [Jin et al. \(2021\)](#) [Xu et al. \(2020\)](#) and [Sun et al. \(2022\)](#). However, these studies focus on the adversarial vulnerabilities of undirected graphs and neural network architectures that do not scale to large-scale knowledge graphs. Research on the adversarial vulnerabilities of representation learning for knowledge graphs has received very little attention. This is precisely the research gap addressed by this thesis.

*This thesis examines the adversarial robustness of representation learning algorithms for knowledge graphs through the design, implementation and evaluation of data poisoning attacks against them.*

Data poisoning attacks aim to degrade the predictive performance of the learned model at inference time by making edits to the input knowledge graph at training time. The focus of this thesis are Knowledge Graph Embedding (KGE) models that are the state-of-the-art algorithms for representation learning on knowledge graphs ([Ali et al., 2021](#); [Rossi et al., 2021](#)). Since the knowledge graphs are often curated automatically from textual sources on the Web or from user generated content, they are incomplete in practice. Thus, an important and state-of-the-art downstream application of KGE models is to predict the missing facts in knowledge graphs ([Nickel et al., 2016a](#)). To investigate the adversarial robustness of KGE models, the thesis focuses on this downstream task of missing link prediction in knowledge graphs. The aim of the research is to design methods that craft facts which can be added to or removed from the original knowledge graph, such that a KGE model trained on the perturbed graph exhibits worse predictive performance than a KGE model trained on the original graph.

### *Research Question*

The research question to be addressed in the thesis is -

*To what extent can the predictive performance of state-of-the-art KGE models be degraded for missing link prediction at inference time by adding or removing systematically crafted triples to the input knowledge graph at training time?*

To address this research question, the thesis addresses three research objectives -

RO<sub>1</sub> Designing an adversarial attack requires a hypothetical attack scenario, which is defined by a threat model of the adversary. The attack threat model specifies the goals, knowledge and capability of the attacker. The first objective of the research is to define this threat model and formulate the problem statement for the corresponding adversarial attacks. This objective

## INTRODUCTION

also involves the identification of major challenges in designing these adversarial attacks, as well as the existing adversarial attacks against KGE models from the literature.

RO2 The second research objective for this thesis is to design and implement novel adversarial attacks to systematically craft deletions or additions to the training dataset that aim to degrade the predictive performance of the learned KGE models. This objective further includes two sub-objectives -

RO2.1 Propose adversarial attacks on KGE models that degrade their predictive performance through *adversarial deletions*.

RO2.2 Propose adversarial attacks on KGE models that degrade their predictive performance through *adversarial additions*.

RO3 The third research objective is to evaluate the effectiveness of the proposed adversarial attacks in degrading the predictive performance of KGE models.

### 1.3 RESEARCH CONTRIBUTIONS

Data poisoning attacks aim to degrade the predictions of the KGE model at inference time by removing or adding adversarial triples to the input knowledge graph at training time. Designing data poisoning attacks against KGE models poses two main challenges. First, to select an adversarial deletion or addition, the impact of a candidate perturbation on the model predictions needs to be measured. However, the naïve approach of re-training a new KGE model for each candidate perturbation is computationally prohibitive. Second, while the search space for adversarial deletions is limited to existing triples in the knowledge graph, it is computationally intractable to enumerate through all the possible candidate additions.

#### *Poisoning via Instance Attribution Methods*

This thesis proposes to use the model-agnostic *instance attribution methods* from Interpretable Machine Learning (Molnar, 2019) to craft adversarial deletions against KGE models. Instance attribution methods identify the training instances that are *influential* to model predictions, that is, deleting these instances from the training data would considerably change the model predictions. These methods are widely used to generate post-hoc explanations for deep neural networks on images (Koh and Liang, 2017; Hanawa et al., 2021) and text (Han et al., 2020; Pezeshkpour et al., 2021). Since the KGE models have relatively shallow neural architectures and the

instance attribution metrics are independent of the black-box models and the input domain, they are a promising approach to estimate the influence of training triples on the KGE model predictions. Yet, despite their promise, they have not been used for KGE models so far. The instance attribution methods are used in this research to address the challenge of measuring the impact of a candidate adversarial deletion on the KGE model predictions.

The research uses three types of instance attribution methods - Instance Similarity that compares the feature representations of target and training triples (Hanawa et al., 2021; Charpiat et al., 2019); Gradient Similarity that compares the gradients of model's loss function due to target and training triples (Hanawa et al., 2021; Charpiat et al., 2019); and Influence Function (Koh and Liang, 2017) which is a principled approach from the robust statistics to estimate the effect of removing a training triple on the KGE model predictions. The most influential training triple identified by the instance attribution metrics is selected as adversarial deletion.

However, data poisoning attacks based on instance attribution methods require a combinatorial search through the candidate perturbations, which does not scale to adversarial additions. Thus, a heuristic method is proposed to use the instance attribution methods for data poisoning by adversarial additions. Given the influential training triple, the triple for adversarial addition is obtained by replacing one of the two entities of the influential triple with the most dissimilar entity in the embedding space. The intuition behind this step is to add a triple that would reduce the influence of the influential triple. This solution overcomes the scalability challenge for adversarial additions by comparing only the entity embeddings to select the replacement. However, this is a heuristic and model-agnostic solution for adversarial additions and might not generate the most effective edits. Further, the method still requires combinatorial search over the neighbourhood triples. Thus, in the next part of the thesis, model-aware adversarial additions, that do not rely on a combinatorial search in the triple space, are proposed.

#### *Poisoning via Relation Inference Patterns*

To degrade the model performance, the inductive abilities of KGE models are exploited to craft the facts for adversarial addition. The inductive abilities of KGE models are expressed through different connectivity patterns like symmetry, inversion and composition between the relations in the knowledge graph (Trouillon et al., 2019; Hamilton, 2020, Chapter 4). These connectivity patterns are referred to as *relation inference patterns*. Specifically, to degrade the KGE model's performance for predicting a target missing fact, this thesis proposes to improve the model's predictive performance on a set of decoy facts.

## INTRODUCTION

A collection of heuristic approaches are proposed to select the decoy facts and craft adversarial additions that can improve the model's predictive performance on decoy facts through different relation inference patterns. The proposed solution addresses the challenge of a large candidate space by breaking down the search into smaller steps over entities and relations. These steps are - (i) determining the adversarial relations; (ii) determining the decoy entities that most likely violate an inference pattern; and (iii) determining the remaining adversarial entities in the inference pattern that are most likely to improve the predictions for decoy triples.

Furthermore, the extent of effectiveness of the attack relying on an inference pattern indicates the KGE model's sensitivity to that pattern. This means that the proposed data poisoning attacks for adversarial additions can also help in understanding the predictive behaviour of KGE models.

In summary, the main contributions of this thesis are -

RC1 Instance attribution based adversarial attacks for adversarial deletions

RC2 Relation Inference pattern based adversarial attacks for adversarial additions

### 1.4 PUBLICATIONS

The research contributions from this thesis have been published as the following articles at international conferences.

1. **Bhardwaj, P.**, Kelleher, J.\*, Costabello, L.\*, & O'Sullivan, D.\* (2021, November). Adversarial Attacks on Knowledge Graph Embeddings via Instance Attribution Methods. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 8225-8239). ([Bhardwaj et al., 2021a](#))
2. **Bhardwaj, P.**, Kelleher, J.\*, Costabello, L.\*, & O'Sullivan, D.\* (2021, August). 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 (Volume 1: Long Papers) (pp. 1875-1888). ([Bhardwaj et al., 2021b](#))
3. **Bhardwaj, P.** (2020, April). Towards Adversarially Robust Knowledge Graph Embeddings. In Proceedings of the AAI Conference on Artificial Intelligence (Vol. 34, No. 10, pp. 13712-13713). ([Bhardwaj, 2020](#))

For publications with multiple authors, the thesis author proposed and designed the research contributions, conducted the evaluation and wrote the

manuscripts. Remaining authors contributed equally (indicated by \* next to their names) through feedback on the research process and the manuscript drafts.

[Bhardwaj et al. \(2021a\)](#) is based on the first research contribution RC 1 of this thesis, which is discussed in Chapter 4 of the thesis. The contribution was published at EMNLP 2021 and it proposes the data poisoning attacks for adversarial deletions using the instance attribution methods. The second research contribution RC 2 of this thesis has been published as [Bhardwaj et al. \(2021b\)](#) at ACL-IJCNLP 2021. This contribution is discussed in Chapter 5 of the thesis and proposes data poisoning attacks for adversarial additions by exploiting the inductive abilities of the KGE models expressed as relation inference patterns. Lastly, the author’s research proposal for this thesis was published at the Doctoral Consortium at AAAI 2020 as [Bhardwaj \(2020\)](#).

## 1.5 THESIS OUTLINE

The next chapter provides the background details for representation learning on knowledge graphs. The pipeline for state-of-the-art KGE models is introduced and relevant components of this pipeline are discussed. The downstream task of missing link prediction and the standard evaluation protocol for this task are also discussed. Chapter 3 formulates the problem statement for designing data poisoning attacks against the KGE models. This chapter defines the attack threat model used for the thesis and identifies the key challenges to be addressed in the design of data poisoning attacks. State-of-the-art poisoning attacks against KGE models and a methodology for evaluating the proposed attacks are also discussed. This chapter addresses the Research Objective RO1.

Chapter 4 proposes the adversarial attacks based on instance attribution methods and evaluates the proposed attacks against state-of-the-art adversarial attacks. In Chapter 5, the adversarial attacks based on relation inference patterns are discussed and evaluated against the state-of-art adversarial attacks. Both Chapters 4 and 5 collectively address Research Objectives RO2 and RO3.

Chapter 6 highlights studies from the wider literature that are related to the research in this thesis. Families of representation learning on graphs with other modalities are discussed, and the literature on understanding the predictive behaviour of KGE models through post-hoc explanations and theoretical or empirical analysis is analyzed. Finally, Chapter 7 concludes the thesis by summarizing the broader impact of the research contributions and discussing the open research problems for building adversarially robust KGE models.

# 2

---

## REPRESENTATION LEARNING FOR KNOWLEDGE GRAPHS

---

Knowledge graphs represent factual knowledge about the world as relationships between concepts and are the de-facto standard for modeling and integrating complex, structured data from diverse sources (Hogan et al., 2021; Noy et al., 2019). Learning and inference on knowledge graphs is critical for intelligent decision making in enterprise applications that aim to utilize the factual world knowledge. Such applications range from question answering systems for search engines and personal assistants, to recommendation systems for social networks and e-commerce, to drug target discovery from biological networks (Noy et al., 2019; Nickel et al., 2016a; Ji et al., 2022).

While a knowledge graph is represented as a collection of facts, the standard Machine Learning (ML) pipeline uses feature vectors as inputs. Thus, integrating knowledge graphs into these pipelines requires a mapping of the concepts and relations in the knowledge graph to feature vectors. Traditionally, these feature vectors were hand-engineered to encode specific structural properties of the graph data (Hamilton, 2020, Chapter 2). But the inflexibility and design expense of hand-engineering has led way to the paradigm of *learning* these representations. *Representation learning* approaches use neural network architectures to learn the mapping from the concepts and relations in the graph to a low-dimensional vector space. This mapping preserves the topological information from the graph as the algebraic operations in the latent feature space. The goal of representation learning algorithms is to optimize the mapping such that the graph structural information to be preserved can be reconstructed from the learned representations or embeddings (Ma and Tang, 2021, Chapter 4).

Based on the structural information they preserve, there are different families of graph representation learning algorithms. Among these, methods based on edge reconstruction are state-of-the-art for representation learning on knowledge graphs. The learned representations of concepts and relations, called *Knowledge*



*Graph Embeddings* (KGE) are used for different downstream tasks, like knowledge base completion (or missing link prediction), triple classification, entity disambiguation, relation extraction, question answering, etc. (Wang et al., 2017; Ali et al., 2021). The research in this thesis focuses on the task of missing link prediction in knowledge graphs using KGE models. This is because the KGE techniques were originally designed for knowledge base completion (Hamilton, 2020, Chapter 4), and because this downstream task is state-of-the-art evaluation for KGE models (Ali et al., 2021; Wang et al., 2017). The relevant background on Knowledge Graph Embeddings for the thesis is covered in this chapter. The KGE models, training strategies, and evaluation metrics included in this research are introduced. Additionally, an overview of the design space for KGE models is discussed in Section 2.4.1 and Section 2.4.2; these design intuitions are relevant for the thesis contributions in Chapter 5.

While the chapter provides the necessary background for the thesis, a comprehensive survey of all the KGE models published and implemented in the literature is beyond the scope of this thesis. Such surveys are available in Ali et al. (2021), Rossi et al. (2021) and Wang et al. (2017). Of these, Ali et al. (2021) is the most recent study that benchmarks the predictive performance of a variety of KGE models for different training strategies and loss functions. Rossi et al. (2021) provides a taxonomy for KGE models and investigates the effect of structural graph properties on the predictive performance. Wang et al. (2017) presents representative techniques for knowledge graph embeddings, methods to incorporate additional graph information, as well as applications for different downstream tasks. Additionally, Chapter 6 of the thesis provides a brief overview of the other families of graph representation learning, which do not scale to multi-relational knowledge graphs. On the other hand, it is also noteworthy that graph representation learning is not the only method for missing link prediction in knowledge graphs. Methods based on observable graph features, rule mining and relation path ranking are discussed in Nickel et al. (2016a) and Chen et al. (2020), but are not the focus for this research.

## 2.1 KNOWLEDGE GRAPHS

A Knowledge Graph (KG) represents the factual knowledge about the world as relationships between entities. It is a directed, labelled graph where the nodes represent the entities of interest, the edges represent the relationships between them, and the edge labels represent the different types of possible relationships. Based on the domain of interest, the entities might represent various real-world concepts like people, objects, organizations, etc. For example, Figure 1 shows a knowledge graph about the financial details of a bank’s customer. The relation-

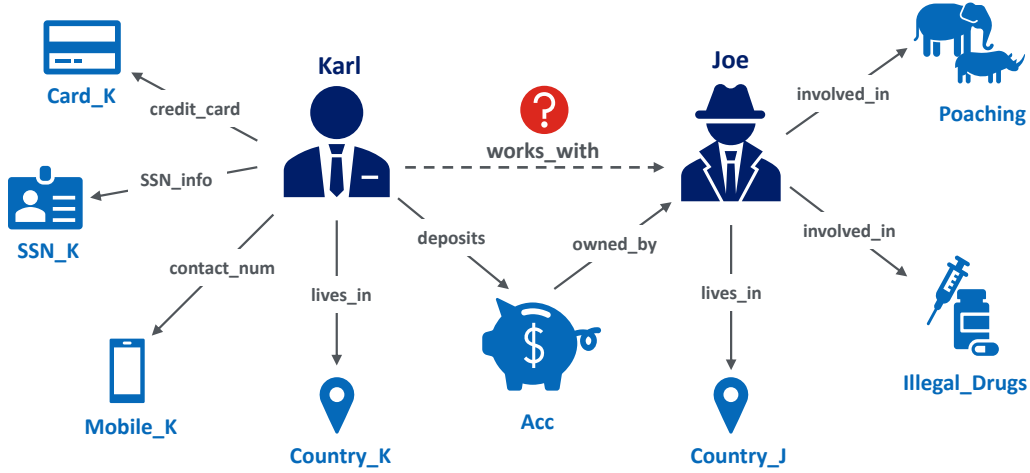


Figure 1: An example Knowledge Graph (KG) of the financial details of a bank’s customer *Karl*. Nodes of the graph represent entities and the edge labels represent types of relationships. The existence of an edge between two entities indicates the relationship between them. Knowledge graph embeddings are used to predict the missing link between *Karl* and *Joe*, given the existing facts in the KG.

ships between the entities can then be expressed as factual statements that convey the real-world knowledge about them. These factual statements of the form  $(subject, relation, object)$  are called triples. In the example knowledge graph, the triple  $(Karl, lives\_in, Country\_K)$  represents the fact that *Karl* lives in the country named *Country\_K*. A knowledge graph then is essentially a collection of facts.

Formally, for a set of entities  $\mathcal{E}$  and a set of relations  $\mathcal{R}$ , a knowledge graph is a collection of triples represented as  $\mathcal{T} := \{t := (s, r, o) \mid s, o \in \mathcal{E} \text{ and } r \in \mathcal{R}\}$ , where  $s, r, o$  represent the subject, relation and object in a triple.

All possible triples from the Cartesian product  $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$  can be grouped as a third order tensor. The existence of a triple in the knowledge graph can then be represented through its corresponding entry in the tensor. This alternate tensor representation of a knowledge graph is called an adjacency tensor (Nickel et al., 2016a). Formally,  $\underline{Y} \in \{0, 1\}^{N_{\mathcal{E}} \times N_{\mathcal{R}} \times N_{\mathcal{E}}}$  is the adjacency tensor with entries

$$y_{ijk} = \begin{cases} 1, & \text{if the triple } (i, j, k) \text{ exists} \\ 0, & \text{otherwise.} \end{cases}$$

In this adjacency tensor,  $y_{ijk} = 1$  indicates the presence of a true triple in the knowledge graph. However, there are different assumptions for the interpretation of  $y_{ijk} = 0$ . Under the Closed World Assumption (CWA), all non-existent triples in the knowledge graph are considered false. In this case, the triple

$(Karl, works\_with, Joe)$ , which is not a part of the example knowledge graph, indicates that *Karl* does not work with *Joe*. On the other hand, under the Open World Assumption (OWA), the non-existing triples are considered as unknown and they can either be true or false. This modelling assumption means that the missing links in the knowledge graph are not necessarily false, and can be inferred by using the existing facts.

## 2.2 KNOWLEDGE GRAPH EMBEDDINGS

Knowledge Graph Embeddings (KGE) encode entities and relations as a low-dimensional continuous vector space  $\theta := \{E, R\}$  where  $E \in \mathbb{R}^k$  is the embedding matrix for entities,  $R \in \mathbb{R}^k$  is the embedding matrix for relations and  $k$  is the embedding dimension. The optimization objective to learn the embeddings aims to reconstruct the relationships between entities as interactions between their latent features. There are many possible ways to model these interactions, which have resulted in different KGE models, characterized by their scoring functions. The KGE scoring function  $f : \mathcal{T} \rightarrow \mathbb{R}$  uses the entity and relation embeddings to assign a score to each triple  $f_t := f(e_s, e_r, e_o) = f(s, r, o)$  where  $e_s, e_o \in E$  and  $e_r \in R$ .

During model training, the embedding matrices are initialized with random values and iteratively updated such that the scores assigned to true (existing) triples in the knowledge graph are higher than the scores for false (non-existing) triples in the knowledge graph. Training this model requires both positive and negative examples, but the knowledge graph only contains positive examples. Thus, negative examples are generated for training by making a Local Closed World Assumption (LCWA), that is, assuming a closed world in the local neighbourhood of an entity. The overall framework for the KGE model training is summarized in Figure 2, and different components of the pipeline are discussed below.

## 2.3 SCORING FUNCTIONS

Table 1 summarizes the scoring functions of state-of-the-art KGE models studied in this research. Representative models have been selected from the three families of KGE models, namely Tensor Factorization models, Geometric models and Deep-learning based models (Rossi et al., 2021; Wang et al., 2017). These representative KGE models have varying inductive abilities (Section 2.4.2) and exhibit multiplicative or additive interactions in the scoring functions (Section 2.4.1).

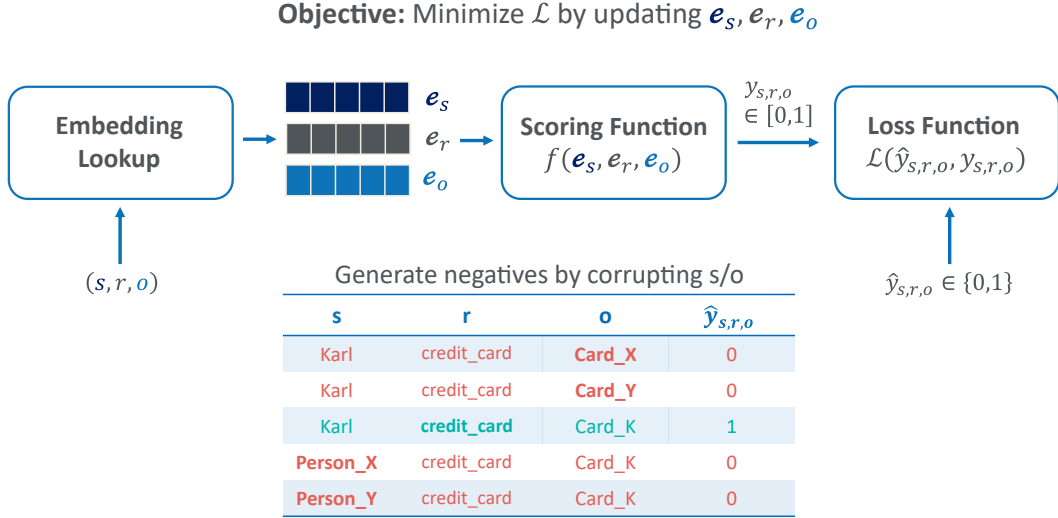


Figure 2: Overall framework for learning Knowledge Graph Embeddings (KGE)

### DistMult

DistMult (Yang et al., 2015) models the interactions between entities as a bilinear product of their embeddings. In general, the bilinear product represents the entity embeddings as vectors, the relation embeddings as matrices and combines them through a matrix multiplication. The DistMult model simplifies this parametrization by constraining the relation matrix to be a diagonal matrix. Thus, the scoring function of the DistMult model is a simple Hadamard product of the subject, relation and object embeddings. The Hadamard product is also called a tri-linear dot product and is denoted as  $\langle \cdot \rangle$  in the expression below.

$$f(s, r, o) = s^T \mathbf{W}_r o = \sum_{i=1}^d s_i \cdot \text{diag}(\mathbf{W}_r)_i \cdot o_i = \langle s, r, o \rangle \quad \text{where } s, r, o \in \mathbb{R}^k$$

Reducing the relation parameters improves the efficiency of DistMult over earlier models like RESCAL (Nickel et al., 2011) which used the bilinear product as a scoring function. However, it reduces the expressivity of the model. For example, for the relation *owns*,  $f(\text{Karl}, \text{owns}, \text{Account})$  should not be equal to  $f(\text{Account}, \text{owns}, \text{Karl})$ , but DistMult cannot model this relationship because of the commutative nature of the dot product. Yet, a recent study of the reproducibility of KGE models in Kadlec et al. (2017) has shown that despite its simplicity, DistMult achieves state-of-the-art predictive performance after proper tuning of the hyperparameters.

Model	Scoring Function
DistMult	$\langle e_s, e_r, e_o \rangle$
ComplEx	$\Re(\langle e_s, e_r, \bar{e}_o \rangle)$
ConvE	$\langle \sigma(\text{vec}(\sigma([\bar{e}_r, \bar{e}_s] * \Omega))W), e_o \rangle$
TransE	$-\ e_s + e_r - e_o\ $

Table 1: Scoring functions  $f_{sro}$  of the KGE models used in this research. For ComplEx,  $e_s, e_r, e_o \in \mathbb{C}^k$ ; for the remaining models  $e_s, e_r, e_o \in \mathbb{R}^k$ . Here,  $\langle \cdot \rangle$  denotes the tri-linear dot product;  $\sigma$  denotes sigmoid activation function,  $*$  denotes 2D convolution with filters  $\Omega$ , and  $W$  denotes additional parameters for ConvE model;  $\bar{\cdot}$  denotes conjugate for complex vectors, and 2D reshaping for real vectors in ConvE model;  $\|\cdot\|$  denotes l-p norm.

### ComplEx

Similar to DistMult, ComplEx (Trouillon et al., 2016) models the latent features as a tensor factorization of the graph adjacency tensor and constrains the relation embedding matrix to be diagonal. However, the latent features are modelled in a complex vector space instead of the real vector space. Here, the embeddings are composed of a real vector component and an imaginary vector component. Unlike real-valued vectors, the bilinear product in complex space is not commutative. This is because the bilinear product for complex numbers is defined as a Hermitian product, which uses the complex conjugate of the object embedding instead of the real-valued object embedding. This acts as a form of topological regularization in the latent space and allows ComplEx to model asymmetric relations like *owns*. Thus, the scoring function for a triple is the real valued component of the Hadamard product between the subject embedding, relation embedding and the complex conjugate of object embedding. Using  $\langle \cdot \rangle$  to denote the Hadamard product and  $\bar{\cdot}$  for the complex conjugate of a vector, the ComplEx scoring function is expressed as follows.

$$f(s, r, o) = \Re(\langle s, r, \bar{o} \rangle) \quad \text{where } s, r, o \in \mathbb{C}^k$$

Some more recent models further improve the expressivity of tensor factorization models. For example, QuatE (Zhang et al., 2019b) represents the latent features in hypercomplex space; and Simple (Kazemi and Poole, 2018) and TuckER (Balazevic et al., 2019b) use further variations of the tensor factorization method like CP decomposition and Tucker decomposition. However, ComplEx remains the state-of-art model for missing link prediction in terms of the predictive performance (Lacroix et al., 2018; Ruffinelli et al., 2020).

*TransE*

Unlike DistMult and ComplEx, TransE (Bordes et al., 2013) belongs to the family of Geometric models or Translational Distance models. These models represent the relationships between entities as geometric transformations in the latent feature space. Specifically, TransE models the object embedding of a triple as a translation from the subject embedding via the relation embedding. In other words, when two entities are related, the embedding of the object is modelled as the embedding of the subject plus the relation embedding. Using  $\|\cdot\|$  to denote the l-p norm of a vector, the TransE scoring function for a triple is expressed as -

$$f(s, r, o) = - \|s + r - o\| \quad \text{where } s, r, o \in \mathbb{R}^k$$

The model is inspired by word embeddings Word2vec (Mikolov et al., 2013) which are learned from free text and coincidentally represent the 1-to-1 relationships between entities as translations in the embedding space. However, the translation operation, explicitly used by TransE, can only model the 1-to-1 relations between entities, which means that TransE cannot handle 1-N, N-1 or N-M relations in the latent space. Despite this limitation, it remains popular because of its computational efficiency and is the representative model for Geometric family of models.

*ConvE*

DistMult, ComplEx and TransE score the triples in knowledge graphs through simple pairwise interactions of the latent features. These shallow neural architectures allow them to scale to real-world knowledge graphs with several million triples. However, the learned features have limited expressivity and increasing the embedding size to improve the expressivity does not scale with the number of entities and relations in the knowledge graph. To overcome these challenges, ConvE (Dettmers et al., 2018) adds a convolutional layer to the neural network architecture. To compute the score of a triple, embeddings of subject and relation are reshaped and concatenated; and the resulting matrix is input to the convolutional layer. In this layer, convolutional filters are applied to capture the complex interactions between the subject and relation features. The output feature maps from the convolutional layer are vectorized and projected back into the k-dimensional embedding space. Finally, these embeddings are combined with the object embeddings using the dot product. Using  $\langle \cdot \rangle$  to denote the dot product,  $\sigma$  to denote the

sigmoid activation function,  $*$  for the 2D convolution operation with filters  $\Omega$ , and  $\bar{\cdot}$  for 2D reshaping of real vectors, the ConvE score for a triple is given by -

$$f(s, r, o) = \langle \sigma(\text{vec}(\sigma([\bar{r}, \bar{s}] * \Omega))W), o \rangle \quad \text{where } s, r, o \in \mathbb{R}^k$$

The convolution operation in ConvE is computationally expensive. Thus, [Dettmers et al. \(2018\)](#) proposed the 1-K training strategy, which scores each  $(s, r)$  pair against all entities simultaneously, instead of scoring the  $(s, r, o)$  triples individually. This enables a single pass through the convolution layer for each  $(s, r)$  pair, thus scaling the model to real-world knowledge graphs.

## 2.4 CHARACTERISTICS OF SCORING FUNCTIONS

### 2.4.1 *Multiplicative vs Additive Interactions*

The scoring functions of KGE models exhibit multiplicative or additive interactions ([Chandrabhas et al., 2018](#)). The multiplicative models score triples through multiplicative interactions of subject, relation and object embeddings. The scoring function for these models can be expressed as  $f_{sro} = e_r^\top \mathcal{F}(e_s, e_o)$  where the function  $\mathcal{F}$  measures the compatibility between the subject and object embeddings and varies across different models within this family. DistMult, ComplEx and ConvE have such interactions. On the other hand, additive models score triples through additive interactions of subject, relation and object embeddings. The scoring function for such models can be expressed as  $f_{sro} = -\|M_r^1(e_s) + e_r - M_r^2(e_o)\|$  where  $e_s, e_o \in \mathbb{R}^{k_\varepsilon}$ ,  $e_r \in \mathbb{R}^{k_\mathcal{R}}$  and  $M_r \in \mathbb{R}^{k_\varepsilon \times k_\mathcal{R}}$  is the projection matrix from entity space  $\mathbb{R}^{k_\varepsilon}$  to relation space  $\mathbb{R}^{k_\mathcal{R}}$ . TransE has additive interactions.

### 2.4.2 *Inductive Abilities of Knowledge Graph Embeddings*

Complementary to multiplicative and additive interactions, another way to characterize the scoring functions of KGE models is based on their ability to represent different logical patterns on the relations ([Trouillon et al., 2019](#); [Hamilton, 2020](#), Chapter 4). This is because the general intuition behind the design of the scoring functions is to capture logical properties between relations from the observed facts in the knowledge graph. These logical properties can then be used to generalize to unseen triples and make downstream inferences about the relationships between entities. For example, the relation `is_owned_by` is inverse of the relation `owns`, and when the fact (`Account, is_owned_by, Karl`) is true, then the fact

(*Karl, owns, Account*) is also true and vice versa. A KGE model that can capture the inversion pattern can thus predict missing facts about *owns* based on the observed facts about *is\_owned\_by*.

In this research, the representational abilities of KGE models are referred to as *relation inference patterns*. The most studied relation inference patterns in the current literature are symmetry, inversion and composition since they occur very frequently in real-world knowledge graphs (Hamilton, 2020; Ali et al., 2021). For a directed graph, the symmetry pattern indicates that if a relation exists in one direction, it also exists in the opposite direction. Similarly, the inversion pattern indicates that the existence of one relation implies the existence of another with opposite direction. The composition pattern indicates that a relation can be expressed as a composition of two or more other relations. These relation inference patterns are discussed in further detail in Chapter 5 where they are used to investigate the adversarial vulnerability of KGE models.

It is noteworthy that the KGE models are latent factor models and the logical patterns are not explicitly encoded in the learning algorithm. Rather, the relation inference patterns are a means to understand the inductive abilities of the models, and reflect the inductive biases that the model might have captured at training time to generalize at inference time.

## 2.5 TRAINING KNOWLEDGE GRAPH EMBEDDINGS

Since the KGs only contain positive triples; to train the KGE model, synthetic negative samples  $t^* \in \mathcal{T}^*$  are generated by replacing the subject or object in the positive triples with other entities in  $\mathcal{E}$ . That is, for each positive triple  $t := (s, r, o)$ , the set of negative samples is  $t^* := \{(s^*, r, o) \cup (s, r, o^*)\}$ .

This approach for negative sample generation relies on the Local Closed World Assumption (LCWA) about the knowledge graph, where the graph is considered only locally complete (Nickel et al., 2016a). This means that if the entity-relation pair  $(s, r)$  has an existing triple  $(s, r, o)$  in the knowledge graph, then any non-existing triple  $(s, r, o^*)$  is treated as false.

The training objective is to learn the embeddings that score positive triples existing in the KG higher than the negative triples generated synthetically. To achieve this, a triple-wise loss function is minimized. This loss function is expressed as

$$\mathcal{L}(t, \theta) := \ell(t, \theta) + \sum_{t^* \in \mathcal{T}^*} \ell(t^*, \theta) \quad \text{where } \theta := \{E, R\}$$



Thus, the optimal parameters  $\hat{\theta}$  learned by the model are defined by  $\hat{\theta} := \arg \min_{\theta} \sum_{t \in \mathcal{T}} \mathcal{L}(t, \theta)$ .

### 2.5.1 Training Strategies

There are different strategies for training the KGE models based on the differences in the way negative triples are generated (Ruffinelli et al., 2020). Using all possible entity corruptions of the training triples can be computationally expensive. Thus, a common approach is to sample some negative triples from the set of all corruptions. Theoretically, the negative corruptions that are false negatives (that is, if the corruption already exists in the graph) should be excluded from the training process. However, in practice, the knowledge graphs are sparse and the likelihood of sampling an actual triple as negative is low. Thus, the additional filter step is omitted for efficiency (Ali et al., 2021). This training strategy is referred to as negative sampling or *NegSamp*, and was proposed by Bordes et al. (2013). An alternative approach for training KGE models was proposed in Lacroix et al. (2018). Instead of sampling from the negative corruptions, the entire set of corruptions is used for training. This approach is called *1vsAll* or *1-N* training strategy. Though it is generally expensive, it is feasible for knowledge graphs where the number of entities is not excessively large (Ruffinelli et al., 2020). Lastly, the approach proposed by Dettmers et al. (2018) is called *KvsAll* or *1-K* training strategy. Instead of generating negative corruptions from individual triples, the approach generates corruptions from entity-relation pairs. In this case, the possible triples from the pair  $(s, r)$  are treated as positive if they exist in the training knowledge graph and negative otherwise.

### 2.5.2 Loss Functions

Several loss functions have also been introduced for training KGE models. Given the model parameters  $\theta$  (that is, embeddings) at a training step, the score for a triple is computed using the scoring functions discussed earlier. The loss functions then compute the error in the scores assigned by the model. This error is computed either from the independent labels for each triple; or by comparing the scores assigned to positive and negative triples; or by comparing the positive triple’s score with a set of negative triples. Based on this, the loss functions can be categorized as pointwise, pairwise and setwise loss functions (Ali et al., 2021; Mohamed et al., 2019).

### *Pointwise Loss Functions*

Pointwise loss functions compute an independent loss term for each triple-label pair. The representative loss function for pointwise losses is Binary Cross Entropy (BCE) loss. It uses the sigmoid activation function on a triple's score to compute the probability of that triple. The error term is then the cross entropy between the resulting probability and the triple's label. Let  $t$  denote a triple (positive or negative example) and  $\hat{y}_t \in \{0, 1\}$  denote the label of the triple. Then, the BCE loss term is expressed as -

$$\mathcal{L}(t, \theta) := -(\hat{y}_t \cdot \log(\sigma(f_t)) + (1 - \hat{y}_t) \cdot \log(1 - \sigma(f_t)))$$

BCE loss thus, frames the learning problem as a binary classification of the triples, where the model outputs are treated as logits. It is suitable for multi-class and multi-label classification and is usually used with the KvsAll training strategy.

### *Pairwise Loss Functions*

The pairwise loss functions compare the scores for positive triples  $t$  with the scores of negative triples  $t^*$ . This is based on the Open World Assumption (OWA) where the negative triples are considered "less positive" instead of negative. The representative loss function for pairwise losses is the margin based ranking loss introduced in [Bordes et al. \(2013\)](#). The loss is applicable to the NegSamp and the 1vsAll training strategies.

Let  $\Delta := f_t - f_{t^*}$  denote the difference between the scores of positive and negative triples. Then, the expression for pairwise hinge loss or the pairwise max-margin loss is given as  $\mathcal{L}(t, \theta) := \max(0, \lambda + \Delta)$ . Here,  $\lambda$  denotes the margin parameter for the difference between the scores.

### *Setwise Loss Functions*

Instead of relying on the scores of individual triples or the pairs of triples, the setwise loss functions compare the positive triple's score with a set of negative triples. The representative function for computing setwise loss is the Cross Entropy (CE) loss. This loss computes the cross entropy between the model's probability distribution and the data distribution of labels. The model distribution is computed as a normalized softmax distribution over the scores assigned by the model. For

a positive triple  $t \in \{(s, r, o)\}$  and all possible object-sided negative corruptions  $t^* \in \{(s, r, o^*)\}$ , the softmax normalization for predicted scores is given by -

$$p(o \mid s, r) = \frac{\exp(f(s, r, o))}{\sum_{o^* \in \mathcal{E}} \exp(f(s, r, o^*))}$$

The data distribution is obtained by normalizing the label values of triples to sum to 1. Finally, the cross entropy between the model and label distribution is -

$$\mathcal{L}(t, \theta) := - \sum_{o^* \in \mathcal{E}} \mathbb{I}[(s, r, o^*) \in \mathcal{T}] \cdot \log(p(o \mid s, r))$$

Here,  $\mathbb{I}$  denotes the indicator function which indicates whether the negative triple already exists in the knowledge graph or not. Since the loss applies a softmax normalization, it is suitable for multi-class and single-label classification problems. It is thus, usually used with NegSamp and 1vsAll training strategies.

## 2.6 MISSING LINK PREDICTION

Large-scale knowledge graphs which serve as background knowledge for knowledge-aware applications are extracted automatically from unstructured textual sources on the Web. Despite advances in information extraction techniques, the extracted graphs are often incomplete and noisy. Thus, predicting the missing facts in knowledge graphs has emerged as an important ML task on knowledge graphs (Nickel et al., 2016a; Rossi et al., 2021). Representation learning on knowledge graphs is the state-of-the-art solution for this task. The learned representations of concepts and relations, called *Knowledge Graph Embeddings* (KGE) are used to score candidate missing facts and select the most plausible one.

The task of knowledge base completion (or missing link prediction) using KGE models is the focus of this research. This task is formulated as an entity ranking problem. Given the entity-relation pairs of the form  $(s, r)$  or  $(r, o)$ , the task is to predict the corresponding object or subject entity. This amounts to answering questions of the form  $(s, r, ?)$  or  $(?, r, o)$ . For example, from the knowledge graph in Figure 1, to predict whether *Karl* works with *Joe*, KGE models are used to answer the query  $(Karl, works\_with, ?)$ . The evaluation protocol for missing link prediction then measures the capability of the link predictor to answer these queries correctly (Ruffinelli et al., 2020; Ali et al., 2021).

### 2.6.1 Evaluation Protocol

The available triples in the knowledge graph are split into training, validation and test subsets. Evaluation of KGE models is performed by training the embeddings on the training subset and using the learned embeddings to correctly predict the triples in the test subset. Similar to the training process, the test subset does not contain negative examples either. Thus, for each test triple  $t = (s, r, o)$ , subject-side corruptions  $t_s^* = (s^*, r, o)$  and object-side corruptions  $t_o^* = (s, r, o^*)$  are generated by replacing the subject/object entity with each entity from  $\mathcal{E}$ . The KGE model’s scoring function is used to predict scores of the original as well as negative triples. The scores are then sorted in descending order and the rank of the correct test entity is determined. These steps provide two ranks for the test triple - one for subject corruptions, and one for object corruptions. A better KGE model assigns better ranks to the actual entity relative to the corruptions. Thus, the entity ranking task evaluates the model’s capability to differentiate between the actual test triples and their corruptions.

While computing the ranks, it is possible that some of the corruptions are valid answers to the query. This happens if the corruptions of the test triple already exist as triples in the training knowledge graph. To avoid the distortion of results due to these false negatives, the filtered evaluation setting was proposed in [Bordes et al. \(2013\)](#). In this setting, negative triples that already exist in the training, validation or test set are filtered out. That is, their scores are ignored while computing the ranks. Depending on the domain of use, either the subject or object or both ranks of the test triple are used to determine the KGE model’s confidence in predicting the missing link.

### 2.6.2 Evaluation Metrics

Given the ranks computed for the test triples, global summary metrics are used to benchmark the predictive performance of different KGE models. Let the set of triples in the test set be denoted by  $\mathcal{T}_{test}$  and the rank of each triple be  $rank(t)$ . Then, following are the state-of-the-art evaluation metrics for missing link prediction with KGE models ([Ali et al., 2021](#)).

*Mean Rank (MR)* is the mean of the ranks predicted by the KGE model. The value of this metric ranges between 1 and  $|\mathcal{E}|$ , and the smaller values indicate

better predictive performance. Due to its sensitivity to outliers, the metric is not used widely. It is defined as -

$$\text{MR} = \frac{1}{|\mathcal{T}_{test}|} \sum_{t \in \mathcal{T}_{test}} \text{rank}(t)$$

*Mean Reciprocal Rank (MRR)* is the mean of the reciprocal values of the predicted ranks. The value ranges between 0 and 1 and the higher values indicate better predictive performance. MRR is defined as -

$$\text{MRR} = \frac{1}{|\mathcal{T}_{test}|} \sum_{t \in \mathcal{T}_{test}} \frac{1}{\text{rank}(t)}$$

*Hits@N* counts the proportion of correct entities that have been ranked less than or equal to N. That is, it measures the fraction of correct entities ranked in the top-N by the KGE model. Based on the measurement granularity of interest, the value of N can be 1, 3, 5, 10 etc. Higher values of Hits@N indicate better predictive performance. The metric is defined as -

$$\text{Hits@N} = \frac{|\{t \in \mathcal{T}_{test} \mid \text{rank}(t) \leq N\}|}{|\mathcal{T}_{test}|}$$

### 2.6.3 Benchmark Datasets

To evaluate the performance of KGE models for missing link prediction, several benchmark datasets have been created over the years (Rossi et al., 2021; Ali et al., 2021). These benchmarks have been sampled from real-world knowledge graphs, instead of being built from scratch. The research in this thesis uses the benchmark datasets WN18RR (Dettmers et al., 2018) and FB15k-237 (Toutanova and Chen, 2015) that have been built from the Wordnet and Freebase knowledge graphs.

#### WordNet

Wordnet<sup>1</sup> is a lexical database of English, designed as a structured representation of the language thesaurus for Computational Linguistics and NLP tasks (Miller, 1995). Semantically similar words that can be used interchangeably are grouped together as concepts called synsets. The word forms (strings of letters representing words) and synsets are represented as nodes of the graph. The labelled edges represent the multiple semantic and lexical relationships between the word forms and their meanings.

<sup>1</sup> <https://wordnet.princeton.edu/>

A subset of the Wordnet knowledge base was extracted by [Bordes et al. \(2014\)](#) to evaluate KGE models for missing link prediction. Triples containing a pre-selected set of relations were selected. Of these, triples containing entities appearing in less than 15 triples were filtered out. This subset is called WN18 and contains 18 relations between 40,943 entities. [Dettmers et al. \(2018\)](#) observed that some facts in the test set of WN18 can be predicted trivially because of an existing inverse relation between the entities in the training set. To avoid this test set leakage, WN18RR was created by removing the facts about relations that have an inverse relation in the knowledge graph. This dataset contains 40,943 entities and 11 relations, and is now the benchmark Wordnet subset to evaluate KGE for missing link prediction.

### *Freebase*

Unlike Wordnet, Freebase ([Bollacker et al., 2008](#)) is an encyclopedic knowledge graph, initially built in the Semantic Web community as a structured representation of general human knowledge. Thus, any real-world concept like person, place, profession or sport can be assigned an identifier and represented as a node in the graph. Factual statements about these entities are then modelled as the *(subject, relation, object)* triples. To support relations between more than two entities, intermediate nodes called Compound Value Type (CVT) nodes are used for reification ([Pellissier Tanon et al., 2016](#); [Rossi et al., 2021](#)). The dataset was built collaboratively with contributions from volunteers on the Web. Though the Freebase API<sup>2</sup> has now been migrated to Wikidata ([Pellissier Tanon et al., 2016](#)), the benchmark subsets extracted from the original dataset continue to be used.

[Bordes et al. \(2013\)](#) extracted a subset of Freebase for KGE model evaluation. Entities with at least 100 mentions and also appearing in Wikilinks database were selected. Facts about these entities were extracted, and the reified relations from the original graph were converted into cliques with binary edges ([Rossi et al., 2021](#), Section 4.3.4). The dataset, called FB15k has 14,951 entities and 1,345 relationships. However, [Toutanova and Chen \(2015\)](#) observed test set leakage for this dataset, that is, entities in some test set triples are connected directly in the training set. A simple model based on observable features could thus achieve state-of-the-art performance for missing link prediction. The study created a subset of FB15k by selecting facts with 401 most frequent relations and filtering out the facts with equivalent or inverse relations. This dataset called FB15k-237 contains 14,541 entities and 237 relations.

It is noteworthy that some of the earliest methods for Knowledge Graph Embeddings were evaluated on the subsets of Wordnet and Freebase ([Bordes et al., 2013](#),

<sup>2</sup> <https://developers.google.com/freebase>

		WN18RR	FB15k-237
Entities		40,559	14,505
Relations		11	237
Triples	Training	86,835	272,115
	Validation	2,824	17,526
	Test	2,924	20,438

Table 2: Statistics for benchmark datasets WN18RR and FB15k-237 used in the thesis.

2014). Though several new benchmarks have been introduced, these remain state-of-the-art datasets for evaluating KGE models. Additionally, the two knowledge bases represent knowledge from two different domains and were constructed using different methodologies. While Wordnet is a lexical ontology that was curated by NLP experts, Freebase is a collection of cross-domain general-purpose facts that were curated by crowdsourcing on the Web. Thus, these two knowledge graphs are suitable for evaluating the research contributions of this thesis. As discussed later in Section 3.6, the thesis uses WN18RR and FB15k-237 for KGE training and evaluation, as well as for the evaluation of the proposed adversarial attacks. Table 2 summarizes the statistics about these datasets.

## 2.7 SUMMARY

This chapter introduced the background concepts for representation learning on knowledge graphs. The overall framework of KGE models, as well as its relevant components were discussed. The task of missing link prediction on knowledge graphs was also introduced. Due to the effectiveness of KGE models for this task, they are now increasingly used in high-impact domains like healthcare and finance, where model outcomes affect people’s lives (Rotmensch et al., 2017; Mohamed et al., 2020; Bonner et al., 2022). In these high-stakes domains there are likely going to be actors that want to cause KGE model failure. However, the security vulnerabilities of KGE models to adversarial manipulation are largely unknown. The next chapter introduces and formulates the problem statement for examining the adversarial vulnerabilities of KGE models.

# 3

---

## PROBLEM FORMULATION

---

Machine Learning (ML) is often contended as critical for building intelligent systems that can operate in complex realistic scenarios (Goodfellow et al., 2016). With the availability of huge volumes of data, ML systems have indeed enabled the automation of many real-world tasks. This is also true for reasoning with knowledge graphs. As discussed in Chen et al. (2020) and Nickel et al. (2016a), latent distributed representations (that is, KGE models) have replaced earlier methods based on logic rules and observable graph features for knowledge graph reasoning. Due to their impressive predictive abilities, the learned representations are increasingly used for user-facing applications in complex safety-critical domains like healthcare and finance (Mohamed et al., 2020; Bonner et al., 2022). This success is due to the ability of representation learning algorithms to encode patterns from the raw input facts in the knowledge graph as latent feature representations for entities and relations.

However, the adaptability of ML models to input data is also a security vulnerability (Barreno et al., 2010; Joseph et al., 2019). Malicious actors that want to manipulate the predictions of KGE models can add or remove carefully and intentionally crafted facts to the input knowledge graph. Since the KGE models are non-interpretable, the impact of malicious edits on their predictions, that is, their failure modes are unknown. This is especially concerning for user-facing applications in high-stakes domains where the cost for model failure is very high. Thus, identifying the security vulnerabilities of KGE models is critical to deploy these models in real-world settings that they are designed for. The security vulnerabilities of ML systems have been investigated in the research field of Adversarial Machine Learning (Biggio and Roli, 2018; Joseph et al., 2019). This chapter introduces the relevant concepts from this field of research and formulates the problem statement for examining the security vulnerabilities of KGE models.



## 3.1 ADVERSARIAL ATTACKS

Machine Learning systems are vulnerable to attacks in adversarial settings, i.e. an adversary can manipulate the results of a learning system by carefully perturbing the inputs (Barreno et al., 2010). Thus, successful deployment of ML systems requires that their security is evaluated during system design. Several studies, Joseph et al. (2019); Biggio and Roli (2018); Liu et al. (2018); Carlini et al. (2019), have been unanimous in their approach to evaluating the security of learning systems. The inspiration is from a well established practice in the field of Computer Security, which is to determine the attacks that can happen on a learning system, then evaluate the resilience of the system by simulating these attacks, and then propose defenses to strengthen the system against these attacks (Joseph et al., 2019). This approach for ML security is based on the underlying principle of proactive security design. According to this design approach, the system designer should anticipate adversarial attacks on the learning system and incorporate defenses against these attacks during system design. The approach is in contrast to a reactive security design approach, in which the defense is proposed in response to an actual attack by an adversary. Thus, designing adversarial attacks against ML models is essential for a proactive and proper security evaluation of these models (Biggio and Roli, 2018).

Adversarial attacks are *empirical* methods to investigate the security of Machine Learning systems<sup>1</sup>. These methods craft systematic and intentional perturbations to the ML model's input data with the aim to perturb the model's predictions. The key design attribute of adversarial attacks is to simulate the data distribution of real-world deployment scenarios where the training and test samples might not belong to the same empirical distribution (Joseph et al., 2019). The stability of the predictive performance of ML models against adversarial attacks indicates their adversarial robustness. Therefore, designing adversarial attacks helps to anticipate the potential security vulnerabilities of an ML model before it is deployed in a real-world (and potentially adversarial) setting.

The aim of adversarial attacks is to perturb the predictions of an ML model by making perturbations to the input data of the model. A taxonomy for adversarial attacks was first proposed in Barreno et al. (2010) and later extended in Huang et al. (2011). The security threats can be categorized based on three different perspectives - the extent of their influence on the learning system, the security violation they cause, and the specificity of the security threats.

---

<sup>1</sup> In contrast to adversarial attacks, *certificates* of adversarial robustness provide theoretical guarantees for the predictions of an ML algorithm for bounds on input perturbations.

## PROBLEM FORMULATION

1. *Influence* – Adversarial attacks can be causative or exploratory. Causative attacks influence the model’s training phase by manipulating the training data. On the other hand, exploratory attacks can only influence the inference phase by manipulating the test data. These two types of attacks are more commonly known as data poisoning and evasion attacks.
2. *Security Violation* – Adversarial attacks can violate the ML system’s integrity by generating false negatives and hence, evading detection without compromising normal system operation; or they can violate availability by generating false positives and hence, compromising the system functionality available to legitimate users; or they can violate privacy by stealing private information from the system.
3. *Specificity* – Adversarial attacks can be targeted to misclassify specific examples, or they can be indiscriminate and misclassify any sample.

In general, an adversarial attack can be conducted at training time or inference time; can have restricted or full access to the learned model’s parameters; and can make noticeable or unnoticeable perturbations to the input data. These different settings can result in a spectrum of attacks for an ML model. However, the adversarial attacks designed for one setting may not be effective for another setting. Hence, for every adversarial attack, an attack threat model provides a specification of the setting for which that attack is designed. This includes the attacker’s goals, their knowledge of the ML system and their capabilities to make perturbations to the system’s input data.

### 3.2 THREAT MODEL FOR ADVERSARIAL ATTACKS

A formal threat model is used to quantify the degree of security needed for a learning system by realistically modeling an adversary (Joseph et al., 2019; Biggio and Roli, 2018; Carlini et al., 2019). It specifies the conditions and assumptions for which the adversarial attack is designed against the learning system. Modeling the threat for an ML system involves defining the adversarial attacker’s goals, their knowledge of the system and their capabilities.

The attack threat model is an essential component of proactive security evaluation, as it enables the system designer to envision different attack scenarios. Corresponding attack strategies can then be proposed and implemented for specific attack scenarios. By specifying the assumptions for adversarial attacks, the attack threat model also determines the computational tractability of designing the attack strategies. It is noteworthy that there is usually a trade-off between the

feasibility of threat models for realistic scenarios and the tractability of adversarial attacks (Joseph et al., 2019; Biggio and Roli, 2018). Specifically, the more restricted an attacker’s knowledge and capabilities are, the more difficult is the design of adversarial attacks for that attack threat model.

Following is the specification of the threat model for designing the adversarial attacks in this research. While this attack threat model is applicable to the proposed attacks in both Chapter 4 and 5, it is further specialized for the attacks proposed in each chapter. Additionally, it closely follows the attack threat model from state-of-the-art adversarial attacks against KGE models (Zhang et al., 2019a; Pezeshkpour et al., 2019; Lawrence et al., 2021), which are discussed in Section 3.4.

### 3.2.1 Attacker’s Goal

The adversarial attacker’s goal is defined using the second and third dimensions of the attack taxonomy. It specifies the security violation, attack specificity and error specificity that an attacker desires to achieve. The attacker may only want to evade detection (integrity attack) or compromise system functionality available to all the users (availability attack) or steal private information from the system (privacy attack). In terms of the attack specificity, the attacker may want to misclassify specific samples (targeted attack) or any sample (indiscriminate attack). Similarly, the attacker may want a sample to be misclassified as a specific class (specific) or as any class other than the true class (generic) (Biggio and Roli, 2018).

This thesis studies the targeted integrity attacks against KGE models for the task of knowledge base completion. The attacker aims to misclassify a specific set of missing triples instead of compromising the overall model performance. Additionally, the attacker aims to misclassify the facts that are predicted True by the victim model, instead of predicting additional missing facts as True. This attack setting is realistic for the scenarios where the KGE models are deployed. For example, if a KGE model is used to predict insurance fraud in a financial network, insurance fraudsters might want to misclassify specific facts that are predicted True by the victim model.

Let the notation  $z := (z_s, z_r, z_o)$  denote the missing *target triple* which is predicted highly plausible by the victim KGE model, that is, assigned a high rank. In this case,  $z_s, z_o$  are the *target entities* and  $z_r$  is the *target relation*. The goal of an adversarial attacker then is to *degrade* the predicted ranks of this triple<sup>2</sup>.

---

<sup>2</sup> This is in contrast to the setting where the attacker would aim to improve the rank of low-ranked triples.

### 3.2.2 *Attacker's Knowledge*

An attack threat model should define the extent of knowledge that an attacker has about the target ML system. This includes the knowledge about system's training data, feature set, learning algorithm and the trained model parameters (Biggio and Roli, 2018). Based on the attacker's knowledge, one can model different attack scenarios – white-box attacks where the attacker has full knowledge, or black-box attacks where the attacker has no knowledge, or gray-box attacks where the attacker has limited knowledge of the target system (Carlini et al., 2019; Barreno et al., 2010).

While black-box settings are more realistic, attacks in white-box settings provide a worst case security evaluation of the ML system. According to Kerckhoffs' principle for Computer Security, the system security should not rely on expectations of secrecy (Joseph et al., 2019). Thus, to ensure reliable vulnerability analysis, this thesis uses a white-box attack setting. It is assumed that the attacker has full knowledge of the victim KGE model architecture as well as access to the learned embeddings.

### 3.2.3 *Attacker's Capability*

A threat model should define the influence that an attacker has on the input data and any reasonable constraints that need to be imposed on the attacker (Biggio and Roli, 2018). The two most common attacks in the attack taxonomy are based on the attack influence - causative attacks that can manipulate both the training and test data, and exploratory attacks that can manipulate test data only. These are more commonly referred to as the training time attacks (or data poisoning attacks), and the test time attacks (or evasion attacks) respectively.

To predict scores for the triples at inference time, state-of-the-art KGE models use only the latent representations learned from the training graph. This means that the perturbations to the graph structure at inference time would not affect the scores predicted for the missing triples. Thus, the evasion attacks at inference time are not realistic for KGE models. This thesis focuses on the data poisoning attacks against KGE models. Data poisoning attacks aim to perturb the predictions of the KGE model at inference time by perturbing the input data at training time. The attacker cannot manipulate the model architecture or the learned embeddings directly; but only through perturbations to the training knowledge graph.

### 3.3 DESIGN SPACE FOR ADVERSARIAL ATTACKS

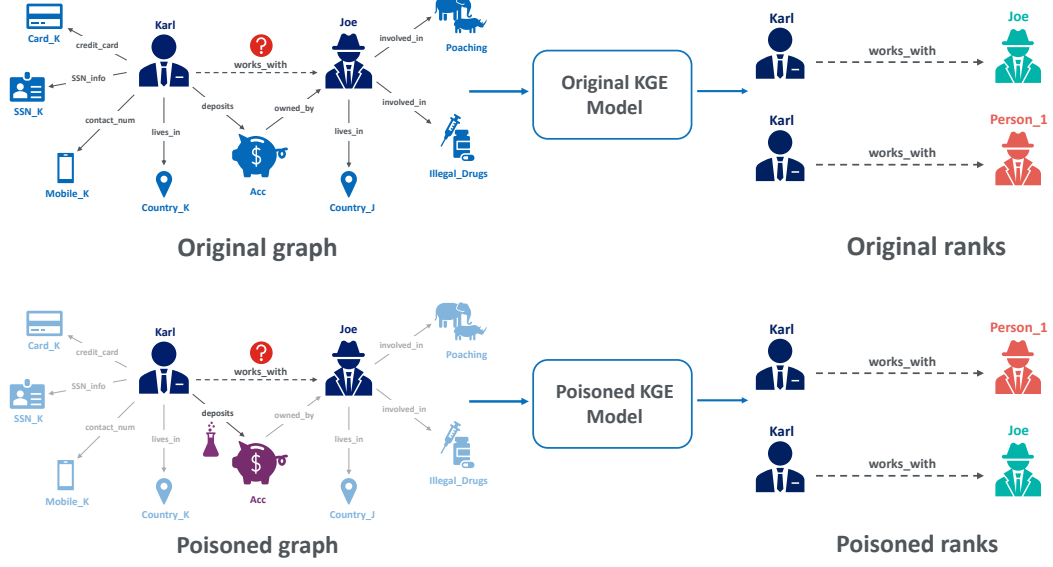


Figure 3: Threat model for the adversarial attacks in this research. The original KGE model assigns a high rank to the target fact ( $Karl, works\_with, Joe$ ). However, deleting the triple ( $Karl, deposits, Acc$ ) from the original knowledge graph, and training a KGE model on the perturbed graph degrades the ranks of the target fact ( $Karl, works\_with, Joe$ ). The change in ranks predicted by the original KGE model and the poisoned KGE model indicates the effectiveness of the adversarial attack.

To summarize, this thesis assumes an adversarial attacker that targets the missing facts which have been assigned high ranks by a KGE model. The attacker aims to design perturbations to the knowledge graph that degrade the target triples' ranks. This means that the target triples' ranks predicted by a KGE model trained on the perturbed graph should be worse than the original model's predictions. Figure 3 illustrates this attack threat model.

### 3.3 DESIGN SPACE FOR ADVERSARIAL ATTACKS

Given the adversarial attacker's goal, knowledge and capability, how can we design an adversarial attack for this threat model? In general, if the victim model's parameters are  $\theta$  and the attacker is capable of perturbing the original dataset  $\mathcal{G}$  to  $\mathcal{G}' \in \Phi$ , then the adversarial goal can be characterized as an objective function  $\mathcal{A}(\mathcal{G}', \theta)$ . Here,  $\Phi$  is the set of valid perturbations on  $\mathcal{G}$ , and the objective function  $\mathcal{A}$  measures the effectiveness of the attack. The optimal strategy to design an adversarial attack then, is to identify the perturbations that would maximize this adversarial objective  $\mathcal{A}$ . That is, the perturbed dataset is given by the solution of the expression  $\arg \max_{\mathcal{G}' \in \Phi} \mathcal{A}(\mathcal{G}', \theta)$  (Biggio and Roli, 2018).

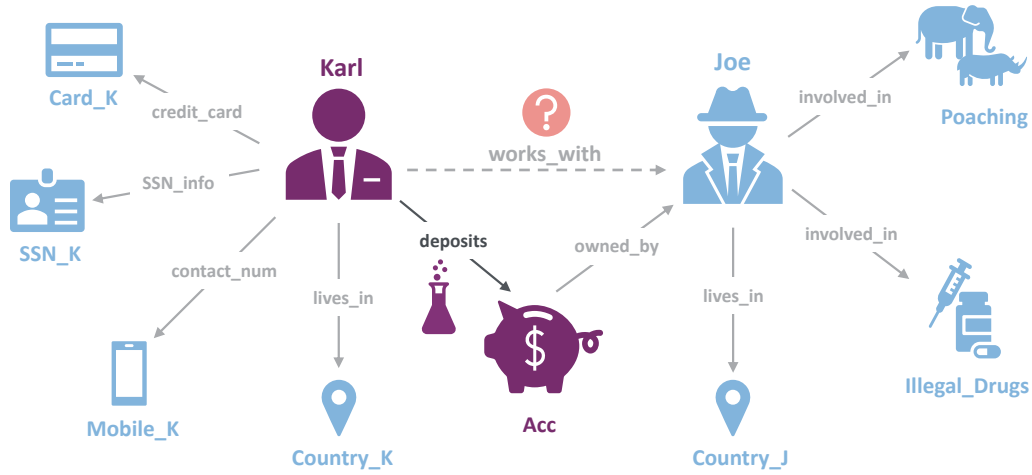
For data poisoning attacks, the perturbations are staged at the training phase. Thus, the victim model parameters  $\theta$  are not static and are constrained to be the optimal solution of model training on the perturbed data. In this setting, the overall attack strategy is formally expressed as a bilevel optimization problem -

$$\hat{\mathcal{G}} := \arg \max_{\mathcal{G}' \in \Phi} \mathcal{A}(\mathcal{G}', \hat{\theta}) \quad s.t. \quad \hat{\theta} := \arg \min_{\theta} \mathcal{L}_{\text{train}}(\theta, \mathcal{G}')$$

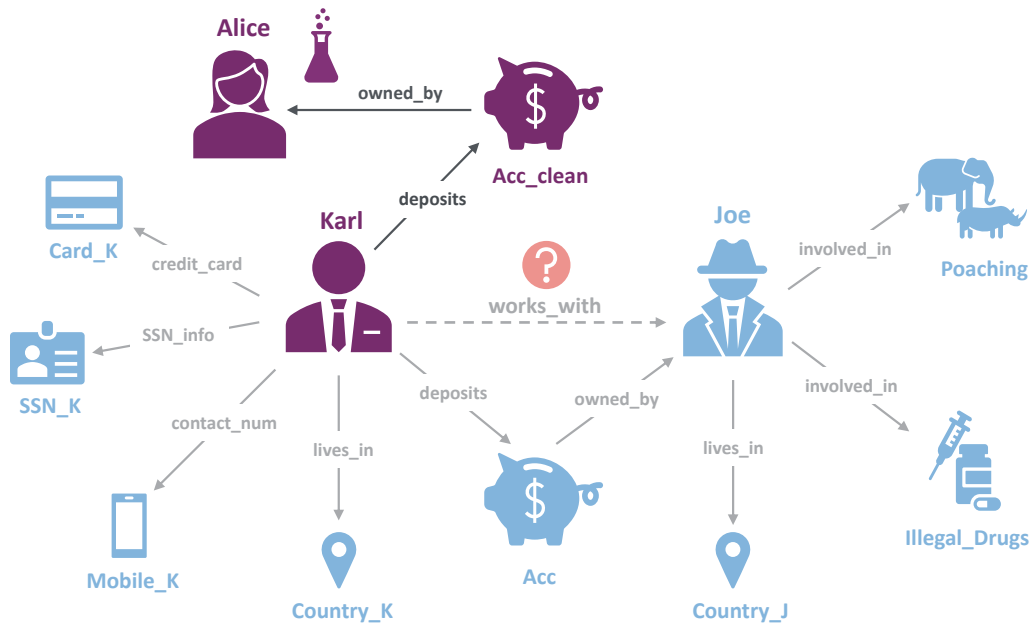
Here, the inner optimization corresponds to learning a model  $\hat{\theta}$  on the perturbed dataset  $\mathcal{G}'$  and the outer optimization problem corresponds to evaluating the impact of these perturbations on the target data through the adversarial objective  $\mathcal{A}$ . When the attacker’s goal is to degrade the predictive performance on specific targets  $\mathcal{Z}$ , the adversarial objective can be defined through a loss function over these targets. That is,  $\mathcal{A}(\mathcal{G}', \hat{\theta}) := \mathcal{L}_{\text{atk}}(\mathcal{Z}, \hat{\theta})$  is used as a measure of the effectiveness of the perturbations on the targets. This loss depends on the perturbed dataset  $\mathcal{G}'$  indirectly through the parameters  $\hat{\theta}$  (Biggio and Roli, 2018; Muñoz González et al., 2017).

In the graph domain, the perturbations can manifest as discrete addition and removal of the entities or triples, or continuous-valued perturbations of the entity features (Günemann, 2022; Zügner et al., 2018). Since the KGE models are trained in a transductive learning setting, the missing links can only be predicted between entities seen during the training. Thus, deletion or addition of entities is not a realistic setting. Additionally, the knowledge graphs in practice do not have pre-existing node features and the embeddings in a KGE model architecture are initialized with random feature values for training (Hamilton, 2020). Thus, perturbations to the entity feature space are also not applicable.

The perturbation space for adversarial attacks in this thesis is the *deletion* or *addition* of triples to the training knowledge graph. This means that the perturbed graph  $\mathcal{G}'$  can be obtained from the original knowledge graph  $\mathcal{G}$  by removing or adding triples to  $\mathcal{G}$ . Additionally, the perturbations are restricted to the neighbourhood of the target triples. The neighbourhood of the target triple  $z := (z_s, z_r, z_o)$  is the set of triples that have the same subject or the same object as the target triple, i.e.  $\mathcal{X} := \{x := (x_s, x_r, x_o) \mid x_s \in \{z_s, z_o\} \vee x_o \in \{z_s, z_o\}\}$ . This restriction is added to make the attack design tractable. However, it is still a realistic restriction as during model training, the embeddings of entities and relation in a triple are updated only due to its neighbourhood triples. Figure 4 illustrates the deletion and addition perturbations for the example knowledge graph from Figure 1.



(a) Adversarial Deletions



(b) Adversarial Additions

Figure 4: An illustration of the perturbation space for adversarial attacks on the example financial knowledge graph in Figure 1. To degrade the KGE model’s predictive performance for target triple  $(Karl, works\_with, Joe)$ , the adversarial attacker can delete the existing triple  $(Karl, deposits, Acc)$ , or they can add the new triples  $(Karl, deposits, Acc\_clean)$  and  $(Acc\_clean, owned\_by, Alice)$ .

### 3.3.1 Challenges

Solving the bilevel optimization for data poisoning attacks is challenging and often intractable, even for the non-graph data modalities (Biggio and Roli, 2018; Muñoz González et al., 2017; Zügner and Günnemann, 2019). The main challenge in this optimization is estimating the impact of a candidate poisoning point on the *solution* of the learning algorithm. A naive approach to measure this impact would be to generate candidate poisoned graphs, train separate models on these candidates and evaluate the adversarial objective for each poisoned model. The candidate perturbation that is most effective for the adversarial objective can then be selected as the adversarial perturbation. However, this approach of re-training a new model for each candidate perturbation is computationally intractable.

Some early studies on data poisoning (for non-graph data modalities) replaced the inner optimization problem with a closed-form solution (Biggio et al., 2012; Muñoz González et al., 2017). This allowed the outer optimization to be solved directly through gradient descent. However, the solution is applicable to very specific families of ML models only, where the equilibrium solution of the inner optimization can be obtained. It is thus, not a practical solution for poisoning neural networks, where these stationarity conditions are difficult to derive.

Another solution explored in Muñoz González et al. (2017); Zügner and Günnemann (2019); Wallace et al. (2021); Franceschi et al. (2019) relies on solving the bilevel optimization through hyperparameter optimization. In this approach, the perturbed dataset is treated as a hyperparameter to be optimized. The hyperparameter is learned by backpropagating the hyper-gradients, that is the gradients of attacker’s loss  $\mathcal{L}_{\text{atk}}(\mathcal{Z}, \hat{\theta})$  with respect to the hyperparameter  $\mathcal{G}'$  in the inner optimization. The hypergradients can be computed by replacing the inner optimization with a set of training iterations performed by the learning algorithm. This solution explicitly saves the parameter updates for the inner optimization, which allows the gradients of the loss in the outer objective to be backpropagated through these updates. However, completely unrolling the entire set of parameter updates is computationally expensive in terms of both time and memory. To resolve this issue, Wallace et al. (2021) and Franceschi et al. (2019) unroll the training process for a fixed number of steps only. Similarly, Zügner and Günnemann (2019) estimates the inner optimization for a fixed number of steps based on a first order approximation of hyper-gradients from the meta-learning literature. On the other hand, Muñoz González et al. (2017) does not store the updates for inner training, but computes the hypergradients directly during backpropagation by reversing the optimization steps.



However, these solutions are not applicable for selecting the optimal perturbations for KGE models. This is because, unlike other neural architectures, the KGE models do not have a differentiable mapping from the graph structure to the embedding domain. For example, the optimization objective for Graph Neural Network (GNN) architectures (Hamilton, 2020) performs convolution operation on the adjacency matrix representation of the graph. This means that a loss function based on the model parameters can be differentiated w.r.t. the adjacency matrix. On the other hand, KGE models use a lookup table to map the graph structure to the embeddings (Chapter 2). Computing the gradients for this lookup operation is not straightforward.

Additionally, the above solutions for data poisoning attacks do not scale to multi-relational knowledge graphs. For images and smaller, undirected graphs, the bilevel formulation optimizes the tensor representation of the image or the adjacency matrix for the graph. The corresponding perturbation space for knowledge graphs would be the adjacency tensor representation of the graph. However, back-propagating the gradients w.r.t. this entire tensor is computationally expensive and in practice, intractable. Thus, for poisoning KGE models in this thesis, the impact of candidate perturbation triples on the attacker’s objective is measured for each candidate triple individually. This in turn introduces a combinatorial search space over the candidate triples to select the adversarial triple.

For adversarial deletions, the combinatorial search space is limited to the existing triples in the knowledge graph. However, it is computationally intractable to enumerate through all the candidate adversarial additions. For example, for every target triple, the possible number of adversarial additions in the neighbourhood of each entity are  $\mathcal{E} \times \mathcal{R}$ . For the benchmark knowledge graphs with thousands of nodes, this search space is of the order of millions for a single target triple.

### 3.3.2 Design Requirements

Based on the above discussion, identifying the optimal perturbed knowledge graph by solving the bilevel attack formulation is not tractable. However, the formal problem specification can be used as an intuitive guide for the design of data poisoning attacks against KGE models. Specifically, for the research in this thesis, the challenges highlighted above are addressed through the following design requirements -

1. *Impact Metric*: The inner and outer optimization in the formal attack specification essentially aims to quantify the impact of a perturbation on the adversarial objective  $\mathcal{A}(\mathcal{G}', \theta)$ . Even without optimizing the bilevel problem,

adversarial perturbations can be identified through a heuristic estimation of this impact. Thus, the design of data poisoning attacks requires an *efficient* method to estimate the impact of a candidate perturbation on the adversarial attack objective  $\mathcal{A}(\mathcal{G}', \theta)$ .

2. *Combinatorial Search*: Given a measure of the perturbation impact, the perturbation with maximum impact needs to be selected. This combinatorial search over the candidate triples is tractable only for the adversarial deletions. Thus, the design of adversarial additions requires an *efficient* method for the combinatorial search over the candidate perturbations.

Along with the discussion in the previous section, these design requirements indicate that attack efficiency is a pre-requisite for the design of effective adversarial attacks against KGE models. Thus, the aim of this research is to design efficient methods for data poisoning attacks that effectively degrade the ranks of the target triples predicted by the KGE model. While the main consideration for attack design is attack effectiveness, designing *effective* data poisoning attacks requires *efficient* solutions.

### 3.4 STATE-OF-THE-ART SOLUTIONS

Recently, some studies have attempted to investigate the adversarial vulnerabilities of representation learning for knowledge graphs. Of these, the studies most closely related to this thesis are [Zhang et al. \(2019a\)](#) that proposes Direct Attacks, [Pezeshkpour et al. \(2019\)](#) that proposes CRIAGE, and [Lawrence et al. \(2021\)](#) that proposes Gradient Rollback. Similar to this thesis, these studies focus on the task of predicting missing links in knowledge graphs using KGE models. Additionally, the attack threat model considered in these studies is similar to the threat model for this research (Section 3.2). The attacker is assumed to have white-box knowledge about the model parameters and aims to degrade the predictive performance by making edits to the training dataset. Thus, the adversarial attacks from these studies are used as baselines to evaluate the attacks proposed in the thesis.

[Zhang et al. \(2019a\)](#) proposed methods for both adversarial deletions and adversarial additions, and used these to investigate the security vulnerabilities of KGE models. [Pezeshkpour et al. \(2019\)](#) also proposed both adversarial deletions and additions, but only used the adversarial additions for poisoning KGE models. The study used adversarial deletions as post-hoc explanations for KGE model predictions. The two tasks of adversarial deletions and post-hoc explanations are equivalent when the explanations for model predictions are counterfactual in na-

ture and the adversarial objective is defined in terms of the model predictions. This is because, both adversarial deletions for the adversarial objective  $\mathcal{A}(\mathcal{G}', \hat{\theta}) := \mathcal{L}_{\text{atk}}(\mathcal{Z}, \hat{\theta})$  and counterfactual explanations for model predictions, aim to identify the facts in the input knowledge graph whose removal would maximally impact the model predictions on the set of target triples  $\mathcal{Z}$ . Thus, both adversarial deletions and additions proposed in Pezeshkpour et al. (2019) are included in this research. Similar to CRIAGE, Lawrence et al. (2021) proposed a method for post-hoc explanations of KGE models. This explanation method is also included for adversarial deletions in this thesis.

### 3.4.1 Adversarial Deletions

As discussed in the previous section, the main design requirement for adversarial deletions is a metric for the impact of a candidate triple’s removal on the adversarial objective. The three methods for adversarial deletions in Zhang et al. (2019a), Pezeshkpour et al. (2019) and Lawrence et al. (2021) specify the adversarial objective in terms of the KGE model’s predicted scores for the target triples. That is, for  $\mathcal{A}(\mathcal{G}', \hat{\theta}) := \mathcal{L}_{\text{atk}}(\mathcal{Z}, \hat{\theta})$ , the attack loss  $\mathcal{L}_{\text{atk}}$  is a function of the KGE model predictions  $f_z$  for target triples  $\mathcal{Z}$ . The methods differ in how the impact of perturbations on the KGE model predictions for target triples is estimated.

Direct Attack (Zhang et al., 2019a) uses the KGE model scores to measure the adversarial candidate’s impact. Given a target triple, Direct Attack first identifies the perturbed embeddings that would degrade the KGE score predicted for the target triple. The attack then ranks the candidate deletions based on the difference in their KGE scores for original and perturbed embeddings. The intuition is to exploit the change in the candidate’s score due to the embedding perturbation as an estimate of the candidate’s impact on perturbing the embedding. Removing a candidate with larger difference in its score would have more impact on perturbing the embedding and in turn the target prediction. However, the experiments in Chapter 4 show that Direct Attack is effective for additive KGE models only.

CRIAGE (Pezeshkpour et al., 2019) estimates the impact of candidate deletions by specializing the Influence Functions (IF) (Koh and Liang, 2017) to KGE models. The influence expression from Koh and Liang (2017) is derived for the specific loss function of Binary Cross Entropy (BCE) on the training triple. The KGE specialized IF expression is computationally more efficient than IF, but it limits the method’s applicability to multiplicative KGE models that are trained using the BCE loss.

On the other hand, Gradient Rollback (Lawrence et al., 2021) estimates the influence of training triples on the learned parameters by keeping track of the gradient

values during the training process. At the start of the training, the influence value for all training triples is initialized to zero. For each training step, the change in parameter values due to a triple is accumulated to the influence value of that triple. After training, the influence of a triple is obtained by subtracting its parameter updates from the model parameters. In other words, the method makes separate copies of the parameter updates due to each training example, and estimates the impact of removing an example by subtracting its copy of parameter updates from the final model parameters. While this method provides theoretical bounds on the influence approximation, it requires additional storage for the separate parameter updates for all training triples. For knowledge graphs with millions of training triples, this overhead becomes expensive. Additionally, to track updates due to each triple, the method can only be applied for a batch size of 1.

In this thesis, the influence of candidate adversarial deletions is estimated through model-agnostic instance attribution methods. Unlike CRIAGE, these methods are applicable for all KGE models. Unlike Gradient Rollback, they do not require additional storage space. Data poisoning attacks using the instance attribution methods are discussed in Chapter 4 of the thesis.

#### 3.4.2 *Adversarial Additions*

As discussed in Section 3.3.2, while the search space for adversarial deletions is limited to candidate triples in the training graph, it is computationally intractable to enumerate through all the possible candidate additions. Adversarial additions against KGE models have been proposed in Direct Attack (Zhang et al., 2019a) and CRIAGE (Pezeshkpour et al., 2019). Both studies follow a similar two-step design approach for adversarial additions.

In the first step, the optimal gradient direction to perturb the latent space is computed. Direct Attack optimizes for the latent representation of the target entity in the target triple. CRIAGE optimizes for a function of the latent representations of entity-relation pair of the unknown adversarial addition. Both methods then determine discrete input perturbations from the perturbed latent space. Direct Attack randomly samples some candidate perturbations and scores them based on the difference of their KGE scores for original and perturbed embeddings. Thus, the attack is similar to the adversarial deletions with Direct Attack, and reduces the search space by random sampling. On the other hand, CRIAGE trains an auto-encoder to decode the entity-relation pair from the perturbed latent space. However, the auto-encoder uses a vanilla neural architecture which is not effective for

generative modelling of the knowledge graphs and supports multiplicative KGE models only .

This thesis addresses the challenge of large candidate space for adversarial additions by breaking down the combinatorial search into three smaller steps over the entities and relations. This is achieved by a novel reformulation of the adversarial objective for poisoning KGE models through adversarial additions. The attacks proposed in Chapter 5 of the thesis exploit the inductive abilities of KGE models to select these adversarial additions.

#### *Adversarial Attacks for Other Threat Models*

The adversarial attacks discussed above consider the same or similar threat model as the attack threat model in Section 3.2, and are thus directly comparable to the contributions of the thesis. However, there is some research that investigates the security vulnerabilities of Machine Learning tasks for knowledge graphs, but focuses on a different attack setting or ML task than the one in this research.

In parallel to the research in this thesis, [Banerjee et al. \(2021\)](#) studies risk aware adversarial attacks against KGE models for link prediction. The study aims to improve the stealth of an adversarial attack by reducing the exposure risk of the attack. It is not included in the evaluation of the proposed attacks, because the research in this thesis focuses on improving the attack effectiveness, instead of the attack stealth.

In another parallel work, [Zhang et al. \(2021\)](#) proposes adversarial attacks against the ML task of cross-lingual knowledge graph alignment. Due to differences in the neural architecture and the ML task, this study has not been included for the evaluation of the proposed attacks in the thesis.

Research in [Raman et al. \(2021\)](#) also generates perturbations to a knowledge graph. These graph perturbations are proposed as a method to understand the stability of KG-augmented NLP tasks like commonsense Question Answering (QA) and item recommendation. The aim of the study is to maximally perturb the graph structure while preserving the model’s predictive performance on the graph. This perturbation goal is the opposite of the goal of adversarial attacks. Thus, the methods proposed in the study are not comparable to the contributions of this thesis.

Additional literature that does not focus on the adversarial robustness of KGE models, but improves our understanding of the model predictions through different methods is discussed in Section 6.2 of Chapter 6.

### 3.5 PROPOSED ATTACK STRATEGY

As discussed in Section 3.2, the goal of the adversarial attacker is to degrade the rank of a highly plausible target triple by deleting or adding triples to the training knowledge graph.

The main requirement for designing data poisoning attacks by adversarial *deletions* is to estimate the impact of a candidate adversarial deletion on the adversarial attack objective  $\mathcal{A}(\mathcal{G}', \hat{\theta})$ . Similar to state-of-the-art solutions, the adversarial objective for deletions in this research is defined in terms of an attack loss  $\mathcal{L}_{\text{atk}}(\mathcal{Z}, \hat{\theta})$  on the target triples  $\mathcal{Z}$ . A proxy for this loss is the score predicted for each target triple using the KGE model’s scoring function  $f_z$  where  $z \in \mathcal{Z}$ . Thus, the aim of the adversarial attacker is to craft adversarial edits that would degrade the predicted score  $f_z$  of the target triple  $z$ .

To measure a candidate perturbation’s impact on the predictions for target triples, this thesis proposes to use the model-agnostic instance attribution methods from Interpretable Machine Learning (Molnar, 2019). These methods identify the training instances that are influential to model predictions, and thus quantify the impact of a candidate perturbation on the predicted score of the target triple. This solution is used to identify the adversarial deletions directly. But it requires a combinatorial search through the candidate perturbations, which does not scale to adversarial additions. Thus, a heuristic method is proposed to select adversarial additions by replacing one of the two entities of the influential triple with the most dissimilar entity in the embedding space. This solution is discussed further in Chapter 4.

It is noteworthy that the adversarial attacker’s aim is to degrade the KGE model’s predictive performance. The predictive performance for a set of target facts is measured using the ranks of the target facts, and the attacker aims to degrade the ranks predicted by the KGE model. As the rank of a target triple relies on the score predicted for the target triple, the solution for adversarial deletions defines the adversarial objective as degrading the predicted scores  $f_z$ . For this adversarial objective, the impact of candidate perturbations on the model predictions needs to be explicitly quantified. Since the impact for all candidates cannot be computed together (as discussed in Section 3.3), a combinatorial search over the candidate space becomes necessary.

The large search space is the main design challenge for adversarial additions. The combinatorial search is made necessary by formulating the adversarial objective in terms of the model predictions  $f_z$ . Thus, the search space can be reduced by re-formulating the adversarial objective  $\mathcal{A}(\mathcal{G}', \hat{\theta})$ . The rank of a highly plausible

target triple can be degraded by improving the rank of a less plausible *decoy triple*, instead of degrading the predicted score of the target triple directly. Thus, for the adversarial additions, this thesis specifies the adversarial objective of degrading the ranks of the target triples as improving the ranks of the decoy triples.

For a target triple  $z := (z_s, z_r, z_o)$ , the decoy triple for degrading the rank on the object side would be  $(z_s, z_r, z'_o)$  and the decoy triple for degrading the rank on subject side would be  $(z'_s, z_r, z_o)$ . The aim of the attacker then, is to select decoy triples from the set of valid synthetic negatives and craft adversarial additions to improve their ranks. The attacker does not add the decoy triple itself as an adversarial edit, rather chooses the adversarial edits that would improve the rank of a missing decoy triple after training. As a measure of the candidate perturbations' impact on the ranks of the decoy triples, the attacker exploits the inductive abilities of KGE models. As discussed in Section 2.4.2, The inductive abilities of KGE models are expressed through the relation inference patterns like symmetry, inversion and composition.

Chapter 5 proposes a collection of heuristic approaches to select the decoy triples and craft adversarial additions that use different inference patterns to improve the model's predictive performance on these decoy triples. To overcome the challenge of large candidate space, the search space is broken down into smaller steps - (i) determine the adversarial relations; (ii) determine the decoy entities that most likely violate an inference pattern; (iii) determine the remaining adversarial entities in the inference pattern that most likely improve the rank of decoy triples.

In summary, the contributions proposed in Chapter 4 and 5 of the thesis differ in their specification of the adversarial attack objective for degrading the predictive performance of KGE models.

### 3.6 EVALUATION PROTOCOL

The research contributions of the thesis are evaluated using state-of-the-art evaluation protocol for data poisoning attacks (Xu et al., 2020; Biggio and Roli, 2018). The aim of this evaluation is to measure the *effectiveness* of the proposed attack strategies in degrading the ranks predicted by the KGE models for the target triples at inference time. Additionally, as discussed in Section 3.3.2, the design of data poisoning attacks requires *efficient* methods to estimate the impact of candidate perturbations and to reduce the combinatorial search space for adversarial additions. Thus, the runtime complexity of the proposed attacks is also analyzed. Furthermore, illustrative examples of the attacks are provided for qualitative analysis and to help understand the predictive behaviour of KGE models.

The predictive performance of KGE models is evaluated using state-of-the-art evaluation protocol for missing link prediction, which was discussed in Section 2.6.1. The knowledge graph is split into training, validation and test facts; a KGE model is trained using the training triples; and evaluated on the test triples. The evaluation task involves an entity ranking procedure. Negative corruptions are generated for the test triple to be predicted, invalid negatives already existing in the training or validation set are filtered out (Bordes et al., 2013), and the test triple is ranked against the negatives using the score predicted by the KGE model. The evaluation metrics reported over the entire test set are MRR and Hits@N.

The experimental setup for evaluating the adversarial attacks against KGE models is as follows. First, a victim KGE model is trained on the original training knowledge graph. This model is used to predict the ranks for the triples in the original test set of the knowledge graph. To assess the attack effectiveness in *degrading* the predictive performance, triples for which the original KGE model has good predictive performance are needed. Thus, a subset of the test set triples that is ranked highly plausible by the original model is selected as target triples. Both Hits@1 and Hits@10 are popular metrics for KGE evaluation, and thus triples with ranks  $\leq 1$  or ranks  $\leq 10$  can be considered as highly plausible. However, the adversarial attacks proposed in Chapter 4 include computationally expensive methods. Thus, to avoid expensive attack computation for a large number of target triples, only the subset with ranks  $\leq 1$  is selected as targets for adversarial deletions. On the other hand, the contributions in Chapter 5 are designed to be efficient (because the main design requirement for adversarial additions is efficient combinatorial search). Thus, the proposed adversarial additions are evaluated using the subset of test set that is ranked  $\leq 10$ .

Given the target triples, adversarial deletions or additions are generated for these target triples using one of the proposed or baseline attack strategies. The original training knowledge graph is then perturbed by removing or adding the adversarial triples to the graph. Finally, a new poisoned KGE model is trained on the perturbed knowledge graph. The hyperparameters for training the poisoned KGE model are the same as the original KGE model. This eliminates the effect of hyperparameter tuning on the predictive performance of the poisoned model. The difference between the predictive performance of the original and poisoned KGE models indicates the effectiveness of different adversarial attacks.

The software implementation for evaluating the research contributions from Chapter 4 is publicly available on the GitHub at <https://github.com/PeruBhardwaj/AttributionAttack>. The implementation for the contributions



from Chapter 5 is available on the GitHub at <https://github.com/PeruBhardwaj/InferenceAttack>.

### 3.6.1 Evaluation Metrics

The *effectiveness* of adversarial attacks is measured as the difference between the ranks of the target triples predicted by the original KGE model and the poisoned KGE model. For the experiments in Chapter 4 and Chapter 5, this is reported as the MRR and the Hits@1 for the target triples due to the original KGE model, as well as the poisoned models obtained by different attack strategies. Lower values of MRR and Hits@1 then indicate the higher effectiveness of an adversarial attack. Additionally, the relative percentage difference between the MRR due to the original and poisoned KGE models is reported. This is measured as

$$\Delta_{MRR} = (MRR_{original} - MRR_{poisoned}) / MRR_{original} * 100$$

The more negative the value for  $\Delta_{MRR}$ , the higher is the attack effectiveness.

As discussed in Section 3.3.2, an efficient attack strategy is a pre-requisite for the design of effective adversarial attacks. Thus, while the primary focus of the evaluation is on attack effectiveness, *efficiency* of the adversarial attacks is also analyzed. The efficiency of different attack strategies is measured as the time taken to generate the adversarial perturbations for all the target triples. Thus, given a set of target triples and the original KGE model, the absolute time taken to identify the complete set of adversarial perturbations for all of these target triples is recorded. This time (in seconds) is reported for the experiments in both Chapter 4 and Chapter 5.

### 3.6.2 Baselines

The proposed adversarial attacks are evaluated against several baselines. These include the random edits to the training knowledge graph, as well as state-of-the-art data poisoning attacks discussed in Section 3.4. The baselines for attack evaluation are as follows.

1. *Random\_n* : For this baseline, random edits are made in the neighbourhood of the target triple. Based on the attack perturbations to be evaluated, the edits are deletions and additions in Chapter 4 and Chapter 5 respectively.
2. *Random\_g* : This baseline attack is obtained by making global random edits to the training knowledge graph. These random edits are not restricted to

the neighbourhood of the target triple, and refer to deletions or additions based on the adversarial attacks to be evaluated in Chapters 4 and 5.

3. *Direct Attack*: The data poisoning attacks from Zhang et al. (2019a) are referred to as Direct-Del and Direct-Add for adversarial deletions and additions respectively. Since the study proposes both adversarial deletions and additions, these attacks are used as baselines in both Chapters 4 and 5.
4. *CRIAGE* : The data poisoning attack from Pezeshkpour et al. (2019) is referred to as CRIAGE. Since the study proposes both adversarial deletions and additions, it is used as a baseline in both Chapters 4 and 5.
5. *Gradient Rollback (GR)*: The method for post-hoc explanations from Lawrence et al. (2021) is referred to as Gradient Rollback (GR). It is used as a baseline for adversarial deletions in Chapter 4.

The evaluation protocols used in Zhang et al. (2019a), Pezeshkpour et al. (2019) and Lawrence et al. (2021) differ with respect to their definition of the target triple’s neighbourhood. Direct Attack (Zhang et al., 2019a) was evaluated for edits in the neighbourhood of the target subject entity only. By contrast, CRIAGE (Pezeshkpour et al., 2019) was proposed and evaluated for edits in the neighbourhood of the object entity only. While both these studies use the same evaluation metrics for attack effectiveness, GR (Lawrence et al., 2021) being a post-hoc explanation method uses different evaluation metrics. Further, all three studies used different KGE models and benchmark datasets for the attack evaluation.

Thus, to ensure a fair evaluation of the research contributions for this thesis, all state-of-the-art methods are re-implemented with the same neighbourhood definition. As discussed in Section 3.3.2, for this thesis, the target triple’s neighbourhood is defined as the set of triples that have the same subject or the same object as the target triple. That is, for the target triple  $z := (z_s, z_r, z_o)$ , the neighbourhood is computed as  $\mathcal{X} := \{x := (x_s, x_r, x_o) \mid x_s \in \{z_s, z_o\} \vee x_o \in \{z_s, z_o\}\}$ . Additionally, the attacks are evaluated against the same KGE models for the same benchmark datasets. Publicly available software implementations for CRIAGE<sup>3</sup> and Gradient Rollback<sup>4</sup> are used. Since an implementation of the Direct Attack is not publicly available, this method is re-implemented from scratch. Further details on the implementation of baselines are discussed in Chapters 4 and 5.

---

<sup>3</sup> <https://github.com/pouyapez/criage>

<sup>4</sup> <https://github.com/carolinlawrence/gradient-rollback>

### 3.6.3 Models and Datasets

The evaluation of the adversarial attacks is conducted against four KGE models – DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), ConvE (Dettmers et al., 2018) and TransE (Bordes et al., 2013). These models are the most widely used models for representation learning on knowledge graphs. Recent studies by Kadlec et al. (2017), Ruffinelli et al. (2020) and Ali et al. (2021) have shown that these models achieve state-of-the-art predictive performance for missing link prediction. Additionally, as discussed in Chapter 2, they have varied inductive abilities and are representative of the different families of KGE models - tensor factorization based models, translation based models and deep learning based models (Rossi et al., 2021; Ali et al., 2021). Thus, these four KGE models are suitable for the evaluation of the research contributions in this thesis.

The proposed adversarial attacks are evaluated on two publicly available<sup>5</sup> knowledge graphs – WN18RR (Dettmers et al., 2018) and FB15k-237 (Toutanova and Chen, 2015). These knowledge graphs are the benchmark datasets for the task of missing link prediction using Knowledge Graph Embeddings. As discussed in Section 2.6.3, they are both subsets of real-world knowledge graphs, namely WordNet and Freebase. Some of the earliest methods for KGE models were evaluated on these knowledge graphs. Despite the introduction of additional benchmark datasets, these remain state-of-the-art datasets for KGE model evaluation. Furthermore, they represent the knowledge from different domains – while WN18RR is derived from a lexical ontology for English, FB15k-237 is a subset of a cross-domain knowledge graph created from crowdsourcing on the Web. Hence, both of these datasets are representative knowledge graphs for the task of missing link prediction and are suitable for evaluating the contributions of this thesis.

## 3.7 SUMMARY

As Knowledge Graph Embeddings are increasingly used in high-stakes domains, examining their security vulnerabilities has become critical. This chapter introduced the problem specification for designing adversarial attacks against KGE models. A brief background on adversarial attacks from the field of Adversarial Machine Learning was discussed, and a taxonomy for defining different attacks was introduced. The threat model for adversarial attacks proposed in this thesis was also introduced. This threat model provides a specification of the adversar-

---

<sup>5</sup> <https://github.com/TimDettmers/ConvE>

## PROBLEM FORMULATION

ial attacker's goals, their knowledge of the victim model and their capability for perturbing the input data.

Based on the attack threat model, the mathematical formulation of data poisoning attacks was introduced. The attacks are expressed as a bilevel optimization problem, where the inner optimization represents the KGE learning process on the poisoned data and the outer optimization evaluates the impact of the poisoned dataset on the adversarial attack objective. Designing methods to solve this optimization problem is challenging. Thus, a set of heuristic design requirements for data poisoning attacks against KGE models was introduced. State-of-the-art poisoning attacks for addressing these requirements were discussed and the adversarial attack strategy proposed in this thesis was defined. Finally, the methodology for evaluating the research contributions of this thesis was discussed.

Building on the problem formulation in this chapter, the next chapter proposes and evaluates the methods for designing adversarial deletions against KGE models for missing link prediction in knowledge graphs.

# 4

---

## POISONING VIA INSTANCE ATTRIBUTION METHODS

---

As discussed in Section 3.3.2, the main design requirement for adversarial deletions against KGE models is a measure for the impact of adversarial candidates on the adversarial objective for the target facts. Let the *target triples* be denoted as  $\mathcal{Z} := \{z := (z_s, z_r, z_o)\}$ . The aim of the attacker is to degrade the ranks predicted for these target triples. Since the rank of a target triple is assigned based on the scores predicted by the KGE model, the adversarial objective can be specified in terms of the KGE model predictions for the target triples. The attack strategy then is to measure the impact of candidate perturbations on the KGE model predictions for target facts and select the candidate perturbation with maximum impact. This chapter investigates the instance attribution metrics from the research field of Interpretable Machine Learning (Molnar, 2019) for designing the adversarial deletions against KGE models. The majority of the contents of this chapter are taken verbatim from the author’s publication Bhardwaj et al. (2021a).

### 4.1 THREAT MODEL

The attack threat model used in this chapter is the same as the threat model defined in Section 3.2. This is also the same threat model as the state-of-the-art data poisoning attacks against KGE models (Pezeshkpour et al., 2019; Zhang et al., 2019a; Lawrence et al., 2021). To ensure reliable vulnerability analysis, the attack setting being considered is a white-box attack setting. In this setting, the attacker has full knowledge of the victim model architecture and access to the learned embeddings. However, they cannot perturb the architecture or the embeddings directly. Rather, the learned model can be affected only through perturbations in the training data. The focus of this chapter are *adversarial deletions*, but a heuristic method to generate *adversarial additions* from these deletions is also proposed. For both types of perturbations, the attacker is restricted to making only one edit in the neighbourhood of the target triple. The neighbourhood of the target triple

$z := (z_s, z_r, z_o)$  is the set of triples that have the same subject or the same object as the target triple, i.e.  $\mathcal{X} := \{x := (x_s, x_r, x_o) \mid x_s \in \{z_s, z_o\} \vee x_o \in \{z_s, z_o\}\}$ .

#### 4.2 INSTANCE ATTRIBUTION METHODS

For adversarial deletions, training triples that have influenced the model’s prediction on the target triple need to be identified. Deleting these influential triples from the training set will likely degrade the prediction on the target triple. Thus, an *influence score*  $\phi(z, x) : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}$  is defined for the pairs of triples  $(z, x) \in \mathcal{T} \times \mathcal{T}$ . This score indicates the influence of training triple  $x$  on the prediction of target triple  $z$ . Larger values of the influence score  $\phi(z, x)$  indicate that removing  $x$  from the training data would cause larger reduction in the predicted score on  $z$ . Thus, the training triple with the highest influence score is selected as the adversarial deletion. Figure 5 illustrates an example - to degrade the prediction for  $(Karl, works\_with, Joe)$ , the influence scores for neighbouring training triples are computed;  $(Karl, deposits, Acc)$  is identified as the most influential triple, and selected as adversarial deletion.

Trivially, the influence score for a training triple  $x$  can be computed by removing  $x$  from the training set and re-training the KGE model on the knowledge graph without  $x$ . However, this is a prohibitively expensive step that requires re-training a new KGE model for every candidate influential triple. The instance attribution methods from the literature on Interpretable Machine Learning (Molnar, 2019) provide a more efficient solution. These methods estimate the influence of training triples on the model predictions without re-training the model. They have been used to provide post-hoc explanations for neural networks for images (Koh and Liang, 2017; Hanawa et al., 2021; Charpiat et al., 2019) and for text (Han et al., 2020; Han and Tsvetkov, 2020; Pezeshkpour et al., 2021). Since these methods are post-hoc in nature, they are agnostic of the model architectures and the input data modalities. Yet, despite their widespread use for neural network explainability, they have not been used for KGE model architectures. Three types of instance attribution methods have been used in the previous literature. These are – Instance Similarity metrics based on the feature representations of the target and the training instances (Hanawa et al., 2021; Charpiat et al., 2019), Gradient Similarity metrics based on the gradients of the model’s loss function for the target and the training instances (Hanawa et al., 2021; Charpiat et al., 2019) and Influence Function (Koh and Liang, 2017) that estimates the effect of leave-one-out training. All of these methods for instance attribution are used to select the most influential training triples as adversarial deletions against KGE models. Besides the Influence

Function proposed in Koh and Liang (2017), prior literature has proposed Relative Influence Function (Barshan et al., 2020) and Fisher Kernel (Khanna et al., 2019) methods for estimating the leave-one-out effect. However, these methods require very expensive computation of the full Hessian matrix or the Fisher Information matrix, and are therefore, not included in this research. The methods used for adversarial attacks against KGE model are discussed below.

#### 4.2.0.1 Instance Similarity

The influence of training triple  $x$  on the prediction of target triple  $z$  is estimated as the similarity of their feature representations. The intuition behind these metrics is to identify the training triples that a KGE model has learnt to be similar to the target triple and thus (might) have influenced the model’s prediction on the target triple. Computing this similarity between triples requires feature vector representations for the triples. On the other hand, standard KGE models provide representations for the entities and relations only and the standard KGE scoring functions assign a scalar score to the triples (Section 2.3). However, it is noteworthy that this scalar value is obtained by reducing over the embedding dimension. For example, in the tri-linear dot product for DistMult (Yang et al., 2015), the embeddings of subject, relation and object are multiplied element-wise and then the scalar score for the triple is obtained by summing over the embedding dimension, i.e.  $f_t := \langle e_s, e_r, e_o \rangle := \sum_{i=1}^k e_{s_i} e_{r_i} e_{o_i}$  where  $k$  is the embedding dimension.

Thus, to obtain feature vector representations for the triples  $f_t : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}^k$ , the state-of-the-art KGE scoring functions are used without reduction over the embedding dimension. For the DistMult model, the triple feature vector is  $f := e_s \circ e_r \circ e_o$  where  $\circ$  is the Hadamard (element-wise) product. Table 3 shows the feature vector scores for different KGE models used in this research.

Given the feature vectors for target triples  $f(z)$  and the feature vectors for training triples  $f(x)$ , the following instance similarity metrics are defined based on the research in Hanawa et al. (2021).

**DOT METRIC:** This metric computes the similarity between the target and training instances as the dot product of their feature vectors. That is,  $\phi_{dot}(z, x) := \langle f(z), f(x) \rangle$

**$\ell_2$  METRIC:** This metric computes similarity as the negative Euclidean distance between the feature vectors of target instance and test instance. That is,  $\phi_{\ell_2}(z, x) := -\|f(z) - f(x)\|_2$

Model	Scoring Function	Feature Vectors
DistMult	$\langle e_s, e_r, e_o \rangle$	$e_s \circ e_r \circ e_o$
Complex	$\Re(\langle e_s, e_r, \bar{e}_o \rangle)$	$\Re(e_s \circ e_r \circ \bar{e}_o)$
ConvE	$\langle (e_s * e_r), e_o \rangle$	$(e_s * e_r) \circ e_o$
TransE	$-\ e_s + e_r - e_o\ _p$	$-(e_s + e_r - e_o)$

Table 3: Scoring functions  $f_{sro}$  and the proposed Triple Feature Vectors  $f_{sro}$  of the KGE models used in this research. For Complex,  $e_s, e_r, e_o \in \mathbb{C}^k$ ; for the remaining models  $e_s, e_r, e_o \in \mathbb{R}^k$ . Here,  $\langle \cdot \rangle$  denotes the tri-linear dot product;  $\circ$  denotes the element-wise Hadamard product;  $\bar{\cdot}$  denotes conjugate for complex vectors;  $\|\cdot\|_p$  denotes l-p norm;  $*$  is the neural architecture in ConvE, i.e.  $e_s * e_r := \sigma(\text{vec}(\sigma([\bar{e}_r, \bar{e}_s] * \Omega))W)$  where  $\sigma$  denotes sigmoid activation,  $*$  denotes 2D convolution;  $\bar{\cdot}$  denotes 2D reshaping of real vectors,  $\Omega$  and  $W$  are the additional model parameters.

**COSINE METRIC:** This metric computes similarity as the dot product between  $\ell_2$  normalized feature vectors of target and test instance, i.e. it ignores the magnitude of the vectors and only relies on the angle between them. That is,  $\phi_{\cos}(z, x) := \cos(f(z), f(x))$

Here, the dot product for two vectors  $\mathbf{a}$  and  $\mathbf{b}$  is denoted as  $\langle \mathbf{a}, \mathbf{b} \rangle := \sum_{i=1}^p a_i b_i$ ; the  $\ell_2$  norm of a vector is denoted as  $\|\mathbf{a}\|_2 := \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle}$ ; and the cos similarity between vectors  $\mathbf{a}$  and  $\mathbf{b}$  is denoted as  $\cos(\mathbf{a}, \mathbf{b}) := \langle \mathbf{a}, \mathbf{b} \rangle / \|\mathbf{a}\|_2 \|\mathbf{b}\|_2$ .

#### 4.2.0.2 Gradient Similarity

The gradient of the loss for triple  $z$  w.r.t. model parameters  $\hat{\theta}$  is denoted as  $\mathbf{g}(z, \hat{\theta}) := \nabla_{\theta} \mathcal{L}(z, \hat{\theta})$ . Gradient similarity metrics compute similarity between the gradients due to target triple  $z$  and the gradients due to training triple  $x$ . The intuition is to assign higher influence to training triples that have similar effect on the model’s parameters as the target triple; and are therefore likely to impact the prediction on target triple (Charpiat et al., 2019). Thus, using the same similarity functions as Instance Similarity, the following three metrics for gradient similarity are defined - Gradient Dot (GD), Gradient  $\ell_2$  (GL) and Gradient Cosine (GC).

$$\text{GD(DOT): } \phi_{\text{GD}}(z, x) := \langle \mathbf{g}(z, \hat{\theta}), \mathbf{g}(x, \hat{\theta}) \rangle$$

$$\text{GL}(\ell_2): \phi_{\text{GL}}(z, x) := - \left\| \mathbf{g}(z, \hat{\theta}) - \mathbf{g}(x, \hat{\theta}) \right\|_2$$

$$\text{GC(COS): } \phi_{\text{GC}}(z, x) := \cos(\mathbf{g}(z, \hat{\theta}), \mathbf{g}(x, \hat{\theta}))$$



## 4.2.0.3 Influence Function

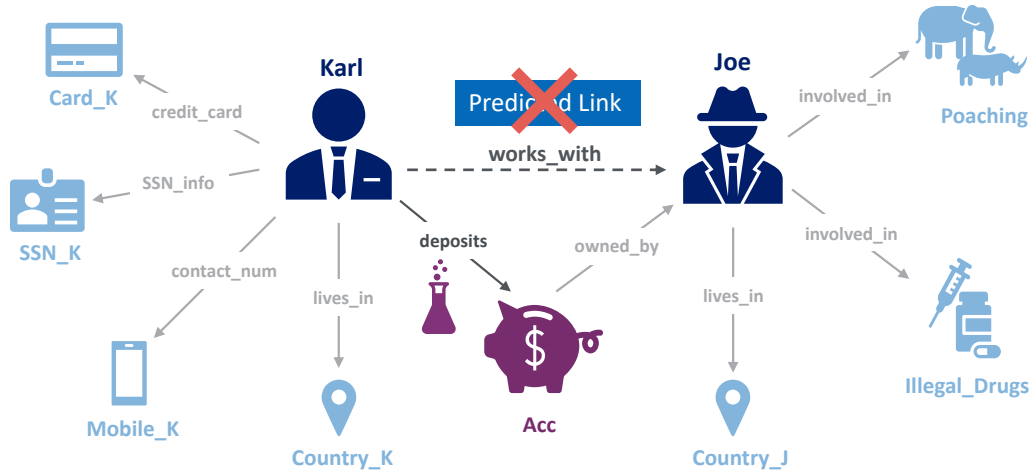
Influence Function (IF) is a classic technique from robust statistics and was introduced to explain the predictions of black-box models in Koh and Liang (2017). To estimate the effect of a training point on a model's predictions, it first approximates the effect of removing the training point on the learned model parameters. To do this, it performs a first order Taylor expansion around the learned parameters  $\hat{\theta}$  at the optimality conditions.

Following the derivation in Koh and Liang (2017), the effect of removing the training triple  $x$  on  $\hat{\theta}$  is given by  $d\hat{\theta}/d\epsilon_i = \mathbf{H}_{\hat{\theta}}^{-1} \mathbf{g}(x, \hat{\theta})$ . Here,  $\mathbf{H}_{\hat{\theta}}$  denotes the Hessian of the loss function  $\mathbf{H}_{\hat{\theta}} := 1/n \sum_{t \in \mathcal{T}} \nabla_{\theta}^2 \mathcal{L}(t, \hat{\theta})$ . Using the chain rule then, the influence of removing  $x$  on the model's prediction at  $z$  is approximated as  $\langle \mathbf{g}(z, \hat{\theta}), d\hat{\theta}/d\epsilon_i \rangle$ . Thus, the influence score using IF is defined as -

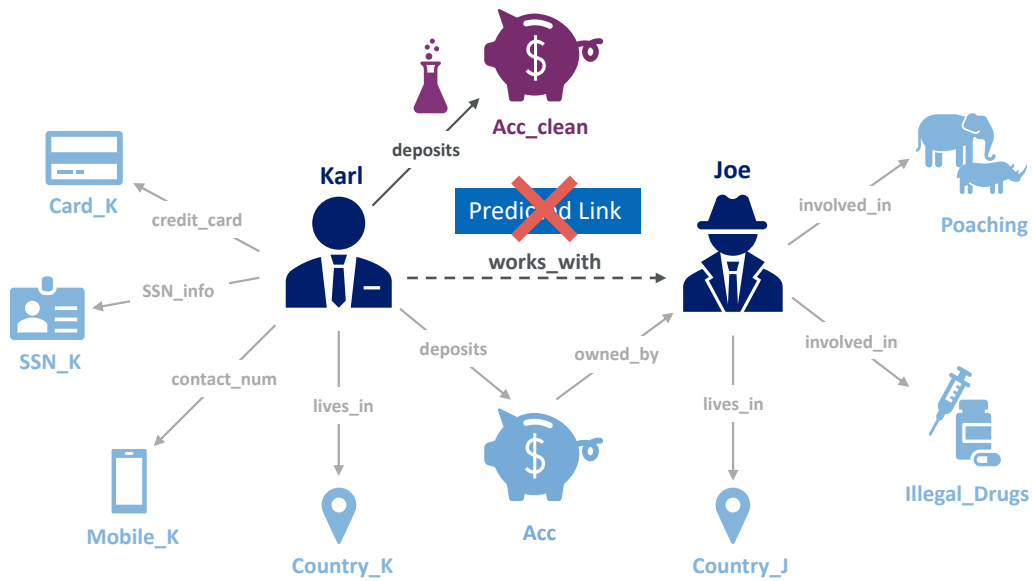
$$\text{IF: } \phi_{\text{IF}}(z, x) := \langle \mathbf{g}(z, \hat{\theta}), \mathbf{H}_{\hat{\theta}}^{-1} \mathbf{g}(x, \hat{\theta}) \rangle$$

Computing the IF for deep learning based models poses two challenges - (i) storing and inverting the Hessian matrix is computationally too expensive for a large number of parameters; (ii) the Hessian is not guaranteed to be positive definite and thus, invertible because deep learning models are non-convex models. Since KGE models are learned from deep neural architectures, both of these challenges hold for KGE models as well. The state-of-the-art solution to address these challenges has been proposed in the original work by Koh and Liang (2017).

To address the first challenge, instead of computing the exact Hessian matrix, a Hessian-vector product (HVP) of the model parameters with the target triple's gradient is estimated. That is, for every target triple  $z$ , the value  $\mathbf{H}_{\hat{\theta}}^{-1} \mathbf{g}(z, \hat{\theta})$  is pre-computed. Then, for each neighbourhood triple  $x$  in the training set, the influence score  $\phi_{\text{IF}}(z, x)$  is computed as the dot product between the pre-computed HVP and the gradient due to  $x$ , that is,  $\mathbf{g}(x, \hat{\theta})$ . Furthermore, as suggested in Koh and Liang (2017), the stochastic estimator LiSSA (Agarwal et al., 2017) is used to estimate the HVP in linear time by sampling the triples from the training data. For the second issue of non-convexity, a "damping" term is added to the Hessian matrix so that it is positive definite and invertible. This term is a hyperparameter that is tuned to ensure that all eigenvalues of the Hessian matrix are positive, i.e. the Hessian matrix is positive definite. Further discussion on the validity of Influence Functions for non-convex settings is available in Koh and Liang (2017).



(a) Adversarial Deletion



(b) Adversarial Addition

Figure 5: Illustrative examples for adversarial attacks based on instance attribution methods. To degrade the prediction for target triple  $(Karl, works\_with, Joe)$ , the attacker can (a) delete the most influential triple  $(Karl, deposits, Acc)$ , or (b) add the triple  $(Karl, deposits, Acc\_clean)$  that reduces the influence of the influential triple by selecting the entity  $Acc\_clean$  that is dissimilar to  $Acc$ .

## 4.3 ADVERSARIAL ADDITIONS

In this attack setting, the adversarial attacker can only *add* triples to the neighbourhood of target triple. The Instance Attribution metrics from the previous section are used to select the training triple  $x := (x_s, x_r, x_o)$  in the neighbourhood of the target triple  $z := (z_s, z_r, z_o)$  that is most influential to the prediction of  $z$ . For brevity, let's assume  $x_s = z_s$ , i.e. the influential and target triples have the same subject. To generate adversarial addition using the influential triple,  $x_o$  is replaced with another entity  $x'_o$ , which is most dissimilar to  $x_o$  in the latent feature space. Since the adversarial triple  $x' := (x_s, x_r, x'_o)$  has the same subject and relation as the influential triple but a different object, it should reduce the influence of the influential triple on the target triple's prediction. This in turn should degrade the model's prediction on the target triple. Consider the knowledge graph in Figure 5 as an example. To degrade the prediction for  $(Karl, works\_with, Joe)$ , the most influential triple identified by an instance attribution method is  $(Karl, deposits, Acc)$ . The influence of this influential triple is reduced by adding  $(Karl, deposits, Acc\_clean)$ , where  $Acc$  has been replaced with  $Acc\_clean$ .

For multiplicative models, the dissimilar entity  $x'_o$  is selected using the cosine similarity between the embedding of  $x_o$  and the embeddings of all entities  $\mathcal{E}$ . For additive models, the  $\ell_2$  similarity between the embedding of  $x_o$  and the embeddings of all entities  $\mathcal{E}$  is used.

## 4.4 EXPERIMENTAL SETUP

As discussed in Section 3.6, the aim of the evaluation is to measure the effectiveness of the proposed attack strategies in degrading the KGE model's predictions on target triples at test time. For this, the state-of-the-art evaluation protocol for data poisoning attacks (Xu et al., 2020) has been used. First, a victim KGE model is trained on the original knowledge graph. Next, adversarial deletions or additions are generated using one of the proposed attack strategies and the original knowledge graph is perturbed. Finally, a new KGE model is trained on the perturbed knowledge graph. The same hyperparameters are used for training the original and poisoned KGE models. The source code implementation of all experiments conducted in this chapter is available at <https://github.com/PeruBhardwaj/AttributionAttack>.

## 4.4.1 Datasets

The proposed adversarial attacks are evaluated on four state-of-the-art KGE models - DistMult, ComplEx, ConvE and TransE; on two publicly available benchmark datasets for link prediction<sup>1</sup>- WN18RR and FB15k-237. To evaluate the predictive performance of original and poisoned KGE models, the standard KGE evaluation protocol is used (Section 2.6.1). In addition, following Bordes et al. (2013) and as is the standard practice, triples from the validation and test set that contain unseen entities are filtered out.

To assess the attack effectiveness in *degrading* performance on triples predicted as True, a set of triples that are predicted as True by the victim model need to be selected. Thus, for this evaluation, a subset of the benchmark test set is selected that has been ranked the best (i.e. ranks=1) by the victim KGE model. If this subset has more than 100 triples, 100 triples are randomly sampled as the *target triples*; otherwise all the triples are used as target triples. This pre-processing step is done to avoid the expensive Hessian inverse computation in the Influence Functions (IF) for a large number of target triples - for each target triple, estimating the Hessian inverse (as an HVP) using the LissA algorithm requires one training epoch.

Table 4 shows the dataset statistics and the number of triples which are ranked best by the different KGE models.

		WN18RR	FB15k-237
Entities		40,559	14,505
Relations		11	237
Training		86,835	272,115
Validation		2,824	17,526
Test		2,924	20,438
Subset with Best Ranks	DistMult	1,109	1,183
	ComplEx	1,198	1,238
	ConvE	1,106	901
	TransE	15	1223

Table 4: Statistics for WN18RR and FB15k-237. Triples from the validation and test set that contained unseen entities were removed to ensure that new entities are not added as adversarial edits. The numbers above (including the number of entities) reflect this filtering.

<sup>1</sup> <https://github.com/TimDettmers/ConvE>

#### 4.4.2 Baselines

The proposed attacks are evaluated against baseline methods based on random edits and the state-of-art data poisoning attacks against KGE models. *Random\_n* is the baseline attack based on random edits that adds or removes a random triple from the neighbourhood of the target triple. *Random\_g* adds or removes a random triple globally and is not restricted to the target’s neighbourhood. *Direct-Del* and *Direct-Add* are the adversarial deletion and addition attacks proposed in Zhang et al. (2019a). *CRIAGE* is the data poisoning attack from Pezeshkpour et al. (2019) and is a baseline for both deletions and additions. *GR* (Gradient Rollback) (Lawrence et al., 2021) uses influence estimation to provide post-hoc explanations for KGE models and can also be used to generate adversarial deletions. Thus, this method is included as a baseline for adversarial deletions only.

The attack evaluations in Zhang et al. (2019a); Pezeshkpour et al. (2019); Lawrence et al. (2021) differ with respect to the definition of their *neighbourhood*. Thus, to ensure fair evaluation for this thesis, all the state-of-art methods are implemented with the same neighbourhood - triples that are linked to the subject or object of the target triple (Section 4.1). For this, the publicly available implementations for *CRIAGE*<sup>2</sup> and Gradient Rollback<sup>3</sup> were used and the *Direct-Del* and *Direct-Add* were implemented by the author herself. Additional details on the implementation of KGE models and baselines; and the computing resources used are available in the Appendix A.1.

### 4.5 EVALUATION OF ATTACK EFFECTIVENESS

For WN<sub>18RR</sub> and FB<sub>15k-237</sub> respectively, Tables 5 and 6 show the degradation in MRR and Hits@1 due to adversarial deletions; and Tables 7 and 8 due to adversarial additions for state-of-the-art KGE models. Different patterns observed in these results are discussed below.

*Note* - For the results reported in the publication associated with this chapter (Bhardwaj et al., 2021a), the implementation of Gradient Rollback (GR) contained a software error. This has been corrected in commit 62a39e7 of the software, and updated results have been included in Tables 5 and 6 of this chapter.

<sup>2</sup> <https://github.com/pouyapez/criage>

<sup>3</sup> <https://github.com/carolinlawrence/gradient-rollback>

	DistMult		Complex		ConvE		TransE	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
<b>Original</b>	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>Baseline Attacks</b>								
Random_n	0.87 (-13%)	0.82	0.85 (-15%)	0.80	0.82 (-18%)	0.79	0.82 (-18%)	0.70
Random_g	0.97	0.95	0.96	0.93	0.99	0.98	0.93	0.87
Direct-Del	0.88	0.77	0.86 (-14%)	0.77	0.71 (-29%)	0.64	0.54 (-46%)	0.37
CRIAGE	0.73 (-27%)	0.66	-	-	Er	Er	-	-
GR	0.92	0.87	0.86	0.81	0.94	0.91	0.84	0.73
<b>Proposed Attacks</b>								
Dot Metric	0.89	0.82	0.85	0.79	0.84 (-16%)	0.80	0.77	0.60
$\ell_2$ Metric	<b>0.25 (-75%)</b>	0.16	0.29 (-71%)	0.20	0.88	0.78	0.62	0.50
Cos Metric	<b>0.25 (-75%)</b>	0.16	0.29 (-71%)	0.20	0.87	0.76	0.56 (-44%)	0.40
GD (dot)	0.28 (-72%)	0.19	0.29	0.21	0.25	0.21	0.71 (-29%)	0.57
GL ( $\ell_2$ )	0.30	0.20	0.28 (-72%)	0.19	0.17 (-83%)	0.12	0.72	0.60
GC (cos)	0.29	0.19	0.29	0.21	0.20	0.16	0.71 (-29%)	0.57
IF	0.28 (-72%)	0.19	0.29 (-71%)	0.20	0.22 (-78%)	0.17	0.71 (-29%)	0.57

Table 5: Reduction in MRR and Hits@1 due to **adversarial deletions on target triples in WN18RR**. Lower values indicate better results; best results for each model are in bold. First block of rows are the baseline attacks with random edits; second block is state-of-the-art attacks; remaining are the proposed attacks. For each block, the *best* reduction in percentage relative to the original MRR; computed as  $(poisoned - original) / original * 100$  is reported.

	DistMult		ComplEx		ConvE		TransE	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
<b>Original</b>	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>Baseline Attacks</b>								
Random_n	0.66 (-34%)	0.52	0.65 (-35%)	0.51	0.62 (-38%)	0.46	0.71 (-29%)	0.56
Random_g	0.68	0.53	0.65 (-35%)	0.51	0.63	0.50	0.75	0.61
Direct-Del	0.59 (-41%)	0.42	0.62 (-38%)	0.47	0.57 (-43%)	0.41	0.62 (-38%)	0.45
CRIAGE	0.62	0.47	-	-	Er	Er	-	-
GR	0.67	0.53	0.63	0.48	0.59	0.42	0.72	0.58
<b>Proposed Attacks</b>								
Dot Metric	0.63	0.47	0.64	0.49	0.60	0.44	0.74	0.62
$\ell_2$ Metric	0.58	0.41	0.56 (-44%)	0.40	0.53 (-47%)	0.35	0.63 (-37%)	0.46
Cos Metric	0.56 (-44%)	0.39	0.57	0.40	0.55	0.38	0.63 (-37%)	0.45
GD (dot)	0.60	0.44	0.60	0.45	0.55 (-45%)	0.37	0.65	0.49
GL ( $\ell_2$ )	0.62	0.45	0.60	0.45	0.56	0.41	0.70	0.58
GC (cos)	0.58 (-42%)	0.42	0.57 (-43%)	0.39	0.57	0.40	0.64 (-36%)	0.48
IF	0.60 (-40%)	0.44	0.60 (-40%)	0.45	0.58 (-42%)	0.43	0.66 (-34%)	0.52

Table 6: Reduction in MRR and Hits@1 due to **adversarial deletions on target triples in FB15k-237**. Lower values indicate better results; best results for each model are in bold. First block of rows are the baseline attacks with random edits; second block is state-of-the-art attacks; remaining are the proposed attacks. For each block, the *best* reduction in percentage relative to the original MRR; computed as  $(poisoned - original) / original * 100$  is reported.

	DistMult		ComplEx		ConvE		TransE	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
<b>Original</b>	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>Baseline Attacks</b>								
Random_n	0.99 (-1%)	0.98	0.97 (-3%)	0.94	0.99 (-1%)	0.98	0.76 (-24%)	0.57
Random_g	0.99 (-1%)	0.97	0.97 (-3%)	0.95	0.99 (-1%)	0.98	0.93	0.87
Direct-Add	0.98 (-2%)	0.96	0.95 (-5%)	0.92	0.99 (-1%)	0.98	0.81 (-19%)	0.67
CRIAGE	0.98 (-2%)	0.97	-	-	Er	Er	-	-
<b>Proposed Attacks</b>								
Dot Metric	0.97	0.93	0.95	0.90	0.95 (-5%)	0.91	0.95	0.90
$\ell_2$ Metric	<b>0.89 (-11%)</b>	0.78	0.88	0.77	0.98	0.96	0.87 (-13%)	0.83
Cos Metric	<b>0.89 (-11%)</b>	0.78	0.87 (-13%)	0.77	0.99	0.98	0.87 (-13%)	0.83
GD (dot)	0.90	0.79	0.89	0.79	0.92	0.85	0.80 (-20%)	0.73
GL ( $\ell_2$ )	<b>0.89 (-11%)</b>	0.79	0.86 (-14%)	0.73	<b>0.88 (-12%)</b>	0.77	0.89	0.83
GC (cos)	0.90	0.80	0.87	0.76	0.91	0.82	0.80 (-20%)	0.73
IF	0.90 (-10%)	0.79	0.89 (-11%)	0.79	0.91 (-8.9%)	0.82	0.77 (-23%)	0.67

Table 7: Reduction in MRR and Hits@1 due to **adversarial additions on target triples in WN18RR**. Lower values indicate better results; best results for each model are in bold. First block of rows are the baseline attacks with random edits; second block is state-of-the-art attacks; remaining are the proposed attacks. For each block, the *best* reduction in percentage relative to the original MRR; computed as  $(poisoned - original) / original * 100$  is reported.



	DistMult		Complex		ConvE		TransE	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
<b>Original</b>	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>Baseline Attacks</b>								
Random_n	0.65 (-34%)	0.50	0.69	0.57	0.61 (-39%)	0.46	0.74	0.62
Random_g	0.66	0.52	0.66 (-34%)	0.52	0.63	0.50	0.73 (-27%)	0.61
Direct-Add	0.64 (-36%)	0.48	0.66 (-34%)	0.52	0.60 (-40%)	0.45	0.72 (-28%)	0.59
CRIAGE	0.66	0.50	-	-	Er	Er	-	-
<b>Proposed Attacks</b>								
Dot Metric	0.67	0.54	0.65	0.50	0.61	0.46	0.74 (-26%)	0.62
$\ell_2$ Metric	0.64	0.50	0.66	0.52	0.59 (-41%)	0.43	0.74 (-26%)	0.62
Cos Metric	0.63 (-37%)	0.49	0.63 (-37%)	0.47	0.60	0.43	0.74 (-26%)	0.61
GD (dot)	0.61 (-39%)	0.45	0.65	0.50	0.62	0.46	0.71 (-29%)	0.58
GL ( $\ell_2$ )	0.63	0.48	0.67	0.53	0.61 (-39%)	0.45	0.74	0.60
GC (cos)	0.62	0.46	0.64 (-36%)	0.49	0.61 (-39%)	0.45	0.71 (-29%)	0.56
IF	0.61 (-39%)	0.45	0.65 (-35%)	0.50	0.58 (-42%)	0.42	0.71 (-29%)	0.58

Table 8: Reduction in MRR and Hits@1 due to **adversarial additions on target triples in FB15k-237**. Lower values indicate better results; best results for each model are in bold. First block of rows are the baseline attacks with random edits; second block is state-of-the-art attacks; remaining are the proposed attacks. For each block, the *best* reduction in percentage relative to the original MRR; computed as  $(poisoned - original) / original * 100$  is reported.

#### 4.5.1 Comparison with Baselines

It is observed that the proposed strategies for adversarial deletions and adversarial additions successfully degrade the predictive performance of KGE models. On the other hand, the state-of-the-art attacks are ineffective or only partially effective. Adversarial deletions from Gradient Rollback perform similar to random baselines; likely because this method estimates the influence of a training triple as the sum of its gradients over the training process. In this way, it does not account for the target triple in the influence estimation. The method is also likely to be effective only for a KGE model that is trained with a batch size of 1 because it needs to track the gradient updates for each triple.

The CRIAGE baseline is only applicable to DistMult and ConvE. However, for the experiments here, the attack ran into `numpy.linalg.LinAlgError: Singular matrix error` for ConvE; because the Hessian matrix computed from the victim model embeddings was non-invertible<sup>4</sup>. For adversarial deletions on DistMult, the baseline works better than random edits but not the proposed attacks<sup>5</sup>. It is also ineffective against adversarial additions.

On the other hand, the Direct-Del is effective on TransE, but not on multiplicative models. This is likely because it estimates the influence of a candidate triple as the *difference* in the triple’s score when the neighbour entity embedding is perturbed. The additive nature of this influence score might make it more suitable for additive models. Furthermore, the Direct-Add works similar to random additions, likely because it uses random down-sampling.

The proposed attacks based on instance attribution methods consistently outperform random baselines for adversarial additions and deletions. One exception to this pattern are adversarial additions against TransE on WN18RR. In this case, no influence metric performs better than random neighbourhood edits, though they are all effective for adversarial deletions. One possible reason is that the TransE model is designed to learn hierarchical relations like `_has_part`. On further investigation, it was found that the target triples ranked highest by the model have such hierarchical relations; and the influential triple for them has the same relation. That is, the triple  $(s_1, \text{\_has\_part}, s)$  is the influential triple for  $(s, \text{\_has\_part}, o)$ . Removing this influential triple breaks the hierarchical link between  $s_1$  and  $s$ ; and degrades TransE predictions on the target. But adding the triple  $(s_2, \text{\_has\_part}, s)$

<sup>4</sup> This issue might be resolved by changing the hyperparameters of the victim KGE model so that the Hessian matrix from the victim embeddings is invertible. But there is no strategic way to make such changes.

<sup>5</sup> Since the influence estimation in CRIAGE uses BCE loss, the attack performance was also compared for DistMult trained with BCE in Appendix A.2, but the results are similar.

Target Relation	Influential Relation
has_part	has_part
synset_domain_topic_of	synset_domain_topic_of
has_part	has_part
synset_domain_topic_of	synset_domain_topic_of
synset_domain_topic_of	synset_domain_topic_of
synset_domain_topic_of	synset_domain_topic_of
instance_hypernym	instance_hypernym
synset_domain_topic_of	synset_domain_topic_of
instance_hypernym	synset_domain_topic_of
synset_domain_topic_of	synset_domain_topic_of
member_meronym	derivationally_related_form
synset_domain_topic_of	synset_domain_topic_of
has_part	has_part
member_meronym	member_meronym
synset_domain_topic_of	synset_domain_topic_of

Table 9: Relations from the target triples and influential triples (adversarial deletions) for the cos metric on WN18RR-TransE. This combination has 15 target triples and the table shows the relations for all of them.

still preserves the hierarchical structure which TransE can use to score the target triple correctly.

The Instance Similarity method cos metric was selected for further analysis. It performs the best of all instance attribution methods for adversarial deletions, but performs worse than random neighbourhood edits for adversarial additions. Table 9 shows the relations in the target triples and the influential triples (i.e. adversarial deletions) selected by cos metric. It is seen that the target triples contain mostly hierarchical relations like `_synset_domain_topic_of` and `_has_part`. Also, the cos metric identifies influential triples with same relations. Since the proposed adversarial additions are only based on modifying the entity in the influential triple, these edits improve the hierarchy structure of the graph instead of breaking it. Thus, these edits perform well for adversarial deletions, but not for additions.

#### 4.5.2 Comparison across Instance Attribution Metrics

Among the different metrics for instance attribution, the IF and Gradient Similarity metrics show similar degradation in predictive performance. This indicates that the computationally expensive Hessian inverse in the IF can be avoided and simpler metrics can identify influential triples with comparable effectiveness. Fur-

thermore,  $\cos$  and  $\ell_2$  based Instance Similarity metrics outperform all other methods for adversarial deletions on DistMult, ComplEx and TransE. This effectiveness of naive metrics indicates the high vulnerability of shallow KGE architectures to data poisoning attacks in practice. In contrast to this, the Input Similarity metrics are less effective in poisoning ConvE, especially significantly on WN18RR. This is likely because the triple feature vectors for ConvE are based on the output from a deeper neural architecture than the Embedding layer alone. Within Instance Similarity metrics, it is observed that the dot metric is not as effective as others. This could be because the dot product does not normalize the triple feature vectors. Thus, training triples with large norms are prioritized over relevant influential triples (Hanawa et al., 2021).

#### 4.5.3 Comparison of Datasets

Among the different datasets, it is noteworthy that the degradation in predictive performance is more significant on WN18RR than on FB15k-237. This is likely due to the sparser graph structure of WN18RR, i.e. there are fewer neighbours per target triple in WN18RR than in FB15k-237. The graph in Figure 6 shows the median number of neighbours of the target triples for WN18RR and FB15k-237. The median value is reported instead of mean because of a large standard deviation in the number of target triple neighbours for FB15k-237.

Based on this graph, the target triple’s neighbourhood for WN18RR is significantly sparser than the neighbourhood for FB15k-237. Since the KGE model predictions are learned from fewer triples for WN18RR, it is also easier to perturb these results with fewer adversarial edits. Thus, removing only 1 neighbour in WN18RR significantly degrades the model’s predictions on the target triple.

On the other hand, because of more neighbours in FB15k-237, the model predictions are likely influenced by a *group* of training triples. Such group effect of training instances on the model parameters has been studied in Koh et al. (2019) and Basu et al. (2020). These methods will be investigated for KGE models on FB15k-237 as part of the future work.

## 4.6 ADDITIONAL EVALUATION

### 4.6.1 Analysis of Runtime Efficiency

In addition to the effectiveness in the degradation of KGE model performance, the runtime efficiency of baseline and proposed attack methods for adversarial

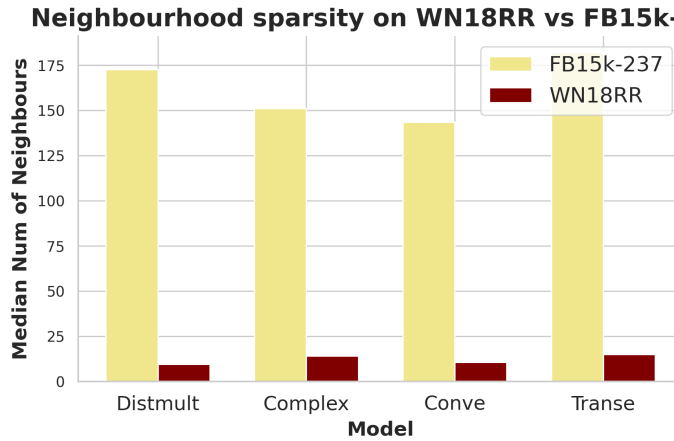


Figure 6: Comparison of the median number of neighbouring triples of target triples from WN18RR and FB15k-237 for DistMult, ComplEx, ConvE and TransE.

deletions is also analyzed. For all the target triples in the test set, the triples in their neighbourhood are pre-computed. Given the original KGE model, the target triples and their neighbourhood, the absolute time taken to select the influential triples for all the target triples is recorded. Table 10 shows the time taken in seconds to select the complete set of adversarial deletions for DistMult model on WN18RR and FB15k-237. For brevity, only the attacks on DistMult model are reported, but the results on other models show similar time scales.

The time scales show that the Instance Similarity metrics (dot metric,  $\ell_2$  metric, cos metric) are more efficient than the state-of-the-art attacks (Direct-Del, CRIAGE and GR). Furthermore, the  $\ell_2$  metric is almost as quick as random triple selection. The efficiency of the Gradient Similarity metrics is also better than or equivalent to CRIAGE and GR. Only the attack method based on IF is much slower than any other method. This is because estimating the Hessian inverse in IF requires one training epoch for every target triple. That is, for the reported experiments, 100 training epochs are done to get the influential triples for 100 target triples. However, based on the results in Section 4.5.2, this expensive computation does not provide improved adversarial deletions, and thus, might be unnecessary to select the influential triples for KGE models.

Comparing the efficiency of attribution attacks for different datasets, the proposed methods are more efficient for the WN18RR dataset than FB15k-237. Since the same number of target triples were used for both datasets (i.e. 100), the difference in efficiency is explained by the difference in neighbourhood sizes of WN18RR and FB15k-237. As Figure 6 from the previous section shows, the neighbourhood of a target triple in FB15k-237 contains more triples than in WN18RR.

		WN18RR	FB15k-237
<b>Baseline Attacks</b>	Random_n	0.024	0.057
	Random_g	0.002	0.002
	Direct-Del	0.407	0.272
	CRIAGE	2.235	75.117
	GR	29.919	174.191
<b>Proposed Attacks</b>	Dot Metric	0.288	0.342
	$\ell_2$ Metric	0.057	0.067
	Cos Metric	0.067	0.148
	GD (dot)	7.354	109.015
	GL ( $\ell_2$ )	8.100	120.659
	GC (cos)	9.478	141.276
	IF	4751.987	4750.404

Table 10: Runtime efficiency of the baseline and proposed adversarial attacks for DistMult on WN18RR and FB15k-237. The absolute time taken in seconds to generate the complete set of influential triples for all target triples is reported. The neighbourhood triples of the target triples were pre-computed. The runtime for GR does not include the time taken to compute the influence map during model training.

Thus, enumerating the neighbours to select the adversarial deletions in WN18RR is more efficient than FB15k-237. The difference in efficiency is more evident for the gradient similarity based attribution methods than the instance similarity based methods. This is because the software implementation for instance similarity methods processes the entire neighbourhood of each triple as a single batch. On the other hand, gradient based methods require enumerating the neighbours to compute the gradient vector for each neighbour individually. Thus, the effect of number of neighbours is more evident for gradient similarity methods than for instance similarity. The reason for similar runtime performance of IF for both datasets is not immediately evident and might be attributed to the hyperparameter setting for computing the IF values. However, for models other than DistMult, there is an observable difference between the IF efficiency for WN18RR and FB15k-237, with better runtime for WN18RR. These values in seconds are – (1305.315 , 5894.426) for ComplEx, (12513.344 , 17433.215) for ConvE and (2613.325 , 7384.060) for TransE.

#### 4.6.2 Qualitative Analysis

This section provides a qualitative analysis of the influential triples selected using the different instance attribution methods for the ComplEx model trained on

Attack	Target and Influential Triples
<b>Dot Metric</b>	<i>dynamical_JJ_1</i> , <i>similar_to</i> , <i>hold-down_NN_1</i> <i>hold-down_NN_1</i> , <i>similar_to</i> , <i>dynamical_JJ_1</i>
$\ell_2$ Metric	<i>departed_NN_1</i> , <i>derivationally_related_form</i> , <i>snuff_it_VB_1</i> <i>snuff_it_VB_1</i> , <i>derivationally_related_form</i> , <i>departed_NN_1</i>
<b>Cos Metric</b>	<i>level_NN_6</i> , <i>derivationally_related_form</i> , <i>tear_down_VB_1</i> <i>tear_down_VB_1</i> , <i>derivationally_related_form</i> , <i>level_NN_6</i>
<b>GD (dot)</b>	<i>personal_identity_NN_1</i> , <i>derivationally_related_form</i> , <i>place_VB_8</i> <i>place_VB_8</i> , <i>derivationally_related_form</i> , <i>personal_identity_NN_1</i>
<b>GL (<math>\ell_2</math>)</b>	<i>dynamitist_NN_1</i> , <i>derivationally_related_form</i> , <i>dynamite_NN_1</i> <i>dynamite_NN_1</i> , <i>derivationally_related_form</i> , <i>dynamitist_NN_1</i>
<b>GC (cos)</b>	<i>departed_NN_1</i> , <i>derivationally_related_form</i> , <i>snuff_it_VB_1</i> <i>snuff_it_VB_1</i> , <i>derivationally_related_form</i> , <i>departed_NN_1</i>
<b>IF</b>	<i>departed_NN_1</i> , <i>derivationally_related_form</i> , <i>snuff_it_VB_1</i> <i>snuff_it_VB_1</i> , <i>derivationally_related_form</i> , <i>departed_NN_1</i>

Table 11: Examples of target triples from WN18RR ComplEx with maximum change in rank due to adversarial deletions, and their corresponding influential triples.

WN18RR and FB15k-237. Tables 11 and 12 show these examples for target triples that exhibit the maximum and minimum change in the ranks respectively due to the adversarial deletions for WN18RR. Since the entities in the benchmark splits are represented as numerical IDs, their string representation are obtained using the *definitions.txt* file. This file is available for download from the original website of WN18RR - <https://everest.hds.utc.fr/doku.php?id=en:smemlj12>. The values for maximum change in ranks due to different attacks for WN18RR are – (35962.5, 6653.5, 39971.5) due to instance similarity metrics, (28011, 37084.5, 36741) due to gradient similarity metrics and 39978.5 due to Influence Function. Tables 13 and 14 focus on the target triples with maximum and minimum change in ranks respectively for FB15k-237. The IDs in the benchmark split are converted into string representations using the browser at <https://freebase.toolforge.org/>. The maximum change in ranks is – (38.5, 31, 80) due to instance similarity metrics, (61.5, 114.5, 149.5) due to gradient similarity metrics and 114.5 due to Influence Function. The minimum change in ranks for both datasets is 0.

Among the target triples with maximum change for WN18RR (Table 11),  $\ell_2$  metric and GC (cos) behave similar to Influence Function. These methods identify the same influential triple (*snuff\_it\_VB\_1* , *derivationally\_related\_form* , *de-*

*parted\_NN\_1*) for the same target triple (*departed\_NN\_1*, *derivationally\_related\_form*, *snuff\_it\_VB\_1*). The entity *departed\_NN\_1* is the noun form to refer to a person who is no longer alive, while the entity *snuff\_it\_VB\_1* refers to the act of passing away from physical life. The two words have similar semantic meaning but different parts-of-speech. Thus, it makes sense for them to be linked by the relation *derivationally\_related\_form*. This indicates that Instance and Gradient Similarity metrics are effective approximations of Influence Function as they are able to identify an influential triple correctly.

Among the triples with no change in ranks for WN18RR (Table 12), GL ( $\ell_2$ ) identifies (*the\_netherlands\_NN\_1*, *derivationally\_related\_form*, *netherlander\_NN\_1*) as the influential triple for (*north\_atlantic\_treaty\_organization\_NN\_1*, *member\_meronym*, *the\_netherlands\_NN\_1*). The target triple indicates that the country Netherlands is a member of NATO. However, the identified influential triple does not provide meaningful attribution for this target prediction, and is thus, ineffective. Some other training triples like (*north\_atlantic\_treaty\_organization\_NN\_1*, *member\_meronym*, *italy\_NN\_1*) or (*europe\_NN\_1*, *has\_part*, *the\_netherlands\_NN\_1*) or (*europe\_NN\_1*, *has\_part*, *italy\_NN\_1*) would be more meaningful influential triples for this target. On the other hand, some influential triples identified by the attribution methods are meaningful, but still ineffective in degrading the target prediction. Consider the GC (cos) attribution method with target triple (*love\_VB\_3*, *derivationally\_related\_form*, *passion\_NN\_6*). While the influential triple (*passion\_NN\_6*, *derivationally\_related\_form*, *love\_VB\_3*) is meaningful, it is not effective. This is because the words ‘love’ and ‘passion’ have multiple meanings which are represented as different entities in the knowledge graph. For example, there are 3 entities *love\_VB\_1*, *love\_VB\_2*, *love\_VB\_3* for love, and 2 entities *passion\_NN\_4*, *passion\_NN\_6* for passion. Furthermore, these entities are linked to each other via *derivationally\_related\_form*. Thus, the target triple’s prediction is influenced by multiple training triples, and deleting one influential triple is insufficient to perturb its prediction.

For FB15k-237 dataset, it can be observed that some of the target triples themselves are unintuitive. For example, from Table 13, the target triple (*University of Louisiana at Monroe*, */common/ topic/ webpage./ common/ webpage/ category*, *Official Website*) indicates that the University of Louisiana (a topic) has a webpage which is the official website for this topic. Similarly, the triple (*The Reader*, */film/ film/ other\_crew./ film/ film\_crew\_gig/ film\_crew\_role*, *Sound Mixer*) indicates that the film *The Reader* has some crewmembers whose job is sound mixing. However, instead of linking the crew member’s role to an entity for the member, the role is linked directly to the movie. Such unintuitive relation links exist in FB15k-237 because



Attack	Target and Influential Triples
<b>Dot Metric</b>	<i>tug_VB_2 , derivationally_related_form , undertaking_NN_1</i> <i>toil_NN_1 , derivationally_related_form , tug_VB_2</i>
$\ell_2$ <b>Metric</b>	<i>genus_alnus_NN_1 , member_meronym , red_alder_NN_1</i> <i>red_alder_NN_1 , hypernym , alder_tree_NN_1</i>
<b>Cos Metric</b>	<i>suddenness_NN_1 , derivationally_related_form , sudden_JJ_1</i> <i>sudden_JJ_1 , derivationally_related_form , suddenness_NN_1</i>
<b>GD (dot)</b>	<i>stock_VB_2 , derivationally_related_form , stock_NN_3</i> <i>stock_NN_3 , hypernym , hold_NN_8</i>
<b>GL (<math>\ell_2</math>)</b>	<i>north_atlantic_treaty_organization_NN_1 , member_meronym ,</i> <i>the_netherlands_NN_1</i> <i>the_netherlands_NN_1 , derivationally_related_form ,</i> <i>netherlander_NN_1</i>
<b>GC (cos)</b>	<i>love_VB_3 , derivationally_related_form , passion_NN_6</i> <i>passion_NN_6 , derivationally_related_form , love_VB_3</i>
<b>IF</b>	<i>suddenness_NN_1 , derivationally_related_form , sudden_JJ_1</i> <i>fast_JJ_1 , also_see , sudden_JJ_1</i>

Table 12: Examples of target triples from WN18RR ComplEx with no change in rank due to adversarial deletions, and their corresponding influential triples.

the original schema of Freebase has Compound Value Type (CVT) entities (Pelissier Tanon et al., 2016) which represent n-ary relations between normal entities. These CVT nodes were removed while preparing the flat graph structure for FB15k and FB15k-237 (Bordes et al., 2013). Thus, the entities linked to CVT nodes in original Freebase data are linked directly in FB15k-237. Further, the multiple hop links between these entities are concatenated into single relations. This leads to semantically incoherent triples, whose predictions are difficult to explain. In addition to the impact of number of neighbours on attack performance (Section 4.5.3), the incoherence of triples in FB15k-237 is also likely responsible for lower effectiveness of the instance attribution methods for this dataset.

On the other hand, looking at target triples without CVT based relations, some influential triples being identified are meaningful, but ineffective in degrading the target triple’s rank. Consider the attacks Cos metric and GL ( $\ell_2$ ) from Table 14. For the target triple (*Harry Potter and the Half Blood Prince* , /film/ film/ story\_by , JK Rowling), the influential triples being selected are (*Harry Potter and the Order of the Phoenix* , /film/ film/ story\_by , JK Rowling) and (*Harry Potter and the Half Blood Prince* , /film/ film/ prequel , *Harry Potter and the Order of the Phoenix*). Both

of these are meaningful influential triples for the target triple, but do not lead to a degradation of rank. This is explained by the training graph for movies *Harry Potter and the Half Blood Prince* and *Harry Potter and the Order of the Phoenix* - both of them share the same acting cast, have the same genres and have been nominated for similar awards. Thus, the prediction for the target triple is not influenced by a single training triple, but multiple ones. This observation motivates the need to define and investigate methods to determine the influence of multiple triples collectively instead of individual triples separately.

#### 4.7 SUMMARY

This chapter proposed the use of instance attribution methods to select adversarial deletions against the KGE models. Additionally, using the influential triples identified from the instance attribution methods, a heuristic approach was proposed to select adversarial additions. The evaluation of the effectiveness of the proposed attacks showed that they outperform the baseline attacks on two benchmark datasets against four KGE models. It was also observed that the attacks are particularly effective when the KGE model relies on fewer training instances to make predictions, i.e. when the input graph is sparse. For the sparse graphs, the proposed attacks are also more efficient than highly connected graphs because they need to enumerate through a smaller set of training triples to select the influential triple. Additionally, shallow neural architectures like DistMult, ComplEx and TransE are vulnerable to simpler attacks based on Instance Similarity than the computationally expensive Influence Function.

While in this chapter, the instance attribution methods were used to select both adversarial deletions and additions, the proposed method for adversarial additions is heuristic in nature. This is because it overcomes the challenge of enumerating through all possible adversarial additions, but still relies on enumerating the neighbourhood triples to select the influential triple. The next chapter proposes an attack strategy for adversarial additions that breaks down the large combinatorial search space of adversarial additions into smaller search steps over the entities and relations in the knowledge graph. To achieve this objective, instead of specifying the adversarial attack objective in terms of the model predictions for the target triples, a reformulation of the attack objective is proposed. This allows the search space to be reduced to three smaller steps.

<b>Attack</b>	<b>Target and Influential Triples</b>		
<b>Dot Metric</b>	<i>The Reader</i>	<i>/film/ film/ other_crew./ film/ film_crew_gig/ film_crew_role</i>	<i>Sound Mixer</i>
	<i>The Devil's Double</i>	<i>/film/ film/ other_crew./ film/ film_crew_gig/ film_crew_role</i>	<i>Sound Mixer</i>
$\ell_2$ Metric	<i>United States of America</i>	<i>/location/ country/ second_level_divisions</i>	<i>Pasco County</i>
	<i>Pasco County</i>	<i>/location/ statistical_region/ rent50_2./ measurement_unit/ dated_money_value/ currency</i>	<i>United States Dollar</i>
<b>Cos Metric</b>	<i>Piper Laurie</i>	<i>/people/ person/ profession</i>	<i>Actor</i>
	<i>Patty Duke</i>	<i>/people/ person/ profession</i>	<i>Actor</i>
<b>GD (dot)</b>	<i>University of Louisiana at Monroe</i>	<i>/common/ topic/ webpage./ common/ webpage/ category</i>	<i>Official Website</i>
	<i>United States of America</i>	<i>/location/ location/ contains</i>	<i>University of Louisiana at Monroe</i>
<b>GL (<math>\ell_2</math>)</b>	<i>University of Louisiana at Monroe</i>	<i>/common/ topic/ webpage./ common/ webpage/ category</i>	<i>Official Website</i>
	<i>University of Louisiana at Monroe</i>	<i>/education/ university/ local_tuition./ measurement_unit/ dated_money_value/ currency</i>	<i>United States Dollar</i>
<b>GC (cos)</b>	<i>Piper Laurie</i>	<i>/people/ person/ profession</i>	<i>Actor</i>
	<i>Piper Laurie</i>	<i>/people/ person/ spouse_s./ people/ marriage/ type_of_union</i>	<i>Marriage</i>
<b>IF</b>	<i>University of Louisiana at Monroe</i>	<i>/common/ topic/ webpage./ common/ webpage/ category</i>	<i>Official Website</i>
	<i>United States of America</i>	<i>/location/ location/ contains</i>	<i>University of Louisiana at Monroe</i>

Table 13: Examples of target triples from FB15k-237 ComplEx with maximum change in rank due to adversarial deletions, and their corresponding influential triples.

Attack		Target and Influential Triples	
<b>Dot Metric</b>	<i>Pompano Beach</i>	<i>/location/ hud_county_place/ place</i>	<i>Pompano Beach</i>
	<i>Pompano Beach</i>	<i>/location/ hud_foreclosure_area/ estimated_number_of_mortgages./ measurement_unit/ dated_integer/ source</i>	<i>United States Department of Housing and Urban Development</i>
$\ell_2$ Metric	<i>Indian Idol</i>	<i>/tv/ tv_program/ country_of_origin</i>	<i>India</i>
	<i>Shatrughan Sinha</i>	<i>/people/ person/ nationality</i>	<i>India</i>
<b>Cos Metric</b>	<i>Harry Potter and the Half Blood Prince</i>	<i>/film/ film/ story_by</i>	<i>JK Rowling</i>
	<i>Harry Potter and the Order of the Phoenix</i>	<i>/film/ film/ story_by</i>	<i>JK Rowling</i>
<b>GD (dot)</b>	<i>Indian Idol</i>	<i>/tv/ tv_program/ country_of_origin</i>	<i>India</i>
	<i>Malayali</i>	<i>/people/ ethnicity/ geographic_distribution</i>	<i>India</i>
<b>GL (<math>\ell_2</math>)</b>	<i>Harry Potter and the Half Blood Prince</i>	<i>/film/ film/ story_by</i>	<i>JK Rowling</i>
	<i>Harry Potter and the Half Blood Prince</i>	<i>/film/ film/ prequel</i>	<i>Harry Potter and the Order of the Phoenix</i>
<b>GC (cos)</b>	<i>Reed College</i>	<i>/education/ educational_institution_campus/ educational_institution</i>	<i>Reed College</i>
	<i>Reed College</i>	<i>/organization/ organization/ headquarters./ location/ mailing_address/ citytown</i>	<i>Portland</i>
<b>IF</b>	<i>Indian Idol</i>	<i>/tv/ tv_program/ country_of_origin</i>	<i>India</i>
	<i>Malayali</i>	<i>/people/ ethnicity/ geographic_distribution</i>	<i>India</i>

Table 14: Examples of target triples from FB15k-237 ComplEx with no change in rank due to adversarial deletions, and their corresponding influential triples.

# 5

---

## POISONING VIA RELATION INFERENCE PATTERNS

---

This chapter proposes methods to select adversarial additions against the KGE models. As discussed in Section 3.3.2, in addition to the design requirement for the impact of candidate perturbations, adversarial additions need to account for the large combinatorial search space. This search space is unavoidable when the adversarial attack objective is defined in terms of the KGE model’s predictions for the target facts. This objective formulation requires the impact of a perturbation to be quantified explicitly. However, the impact of a perturbation on the ranks predicted for target facts can also be defined implicitly through the impact on synthetic negative triples against which the target fact is ranked. This chapter formulates the adversarial attack objective of degrading the predicted ranks of target facts as the task of improving the predicted ranks of decoy triples.

The notation  $(s, r, o)$  is used for the *target triple*; in this case,  $s, o$  are the *target entities* and  $r$  is the *target relation*. The rank of a highly plausible target triple can be degraded by improving the rank of less plausible *decoy triples*. For a target triple  $(s, r, o)$ , the decoy triple for degrading the rank on object side would be  $(s, r, o')$  and the decoy triple for degrading the rank on subject side would be  $(s', r, o)$ . Thus, the aim of the adversarial attacker in this chapter, is to select decoy triples from the set of valid synthetic negatives and craft *adversarial additions* to improve their ranks. The attacker does not add the decoy triple itself as an adversarial edit, rather chooses the adversarial edits that would improve the rank of a missing decoy triple through an inference pattern. The majority of contents in this chapter are taken verbatim from the author’s publication [Bhardwaj et al. \(2021b\)](#).

### 5.1 THREAT MODEL

The threat model for this chapter is the same as the threat model defined in Section 3.2 and the prior state-of-the-art attacks against KGE models ([Pezeshkpour et al., 2019](#); [Zhang et al., 2019a](#)). The attacks are designed for a white-box attack

setting. In this attack setting, the attacker has full knowledge of the target KGE model and the training dataset (Joseph et al., 2019). However, they cannot manipulate the model architecture or the learned embeddings directly; but only through *addition* of triples to the training knowledge graph. Additionally, the attacker is restricted to making edits only in the neighbourhood of target entities. They are also restricted to 1 decoy triple for each entity of the target triple. Furthermore, because of the use of filtered settings for KGE evaluation (Bordes et al., 2013), the attacker cannot add the decoy triple itself to the training data (which intuitively would be a way to improve the decoy triple’s rank).

## 5.2 RELATION INFERENCE PATTERNS

As discussed in Section 2.4.2, the general intuition behind the design of the scoring functions of KGE models is to capture logical properties between relations from the observed facts in the knowledge graph. These logical properties or *relation inference patterns* can then be used to make downstream inferences about entities and relations. For example, the relation `is_owned_by` is inverse of the relation `owns`, and when the fact  $(Account, is\_owned\_by, Karl)$  is true, then the fact  $(Karl, owns, Account)$  is also true and vice versa. A model that can capture inversion pattern can thus predict missing facts about `owns` based on observed facts about `is_owned_by`. The most studied inference patterns in the current literature are symmetry, inversion and composition since they occur very frequently in real-world knowledge graphs (Abboud et al., 2020; Ali et al., 2021). In this chapter, the relation inference patterns are used to improve the ranks of the decoy triples and in turn, degrade the ranks of the target triples.

Since the inference patterns on the knowledge graph specify a logic property between the relations, they can be expressed as Horn Clauses which is a subset of the First Order Logic (FOL) formulae. For example, a property represented in the form  $\forall x, y : (x, owns, y) \Rightarrow (y, is\_owned\_by, x)$  means that two entities linked by the relation `owns` are also likely to be linked by the inverse relation `is_owned_by`. In this expression, the right hand side of the implication  $\Rightarrow$  is referred to as the *head* and the left hand side as the *body* of the clause. Using such expressions, the three inference patterns used in this research are defined as follows.

**Definition 5.2.1.** The *symmetry* pattern  $\mathcal{P}_s$  is expressed as  $\forall x, y : (x, r, y) \Rightarrow (y, r, x)$ . Here, the relation `r` is symmetric relation.

**Definition 5.2.2.** The *inversion* pattern  $\mathcal{P}_i$  is expressed as  $\forall x, y : (x, r_i, y) \Rightarrow (y, r, x)$ . Here, the relations `ri` and `r` are inverse of each other.

**Definition 5.2.3.** The *composition* pattern  $\mathcal{P}_c$  is expressed as  $\forall x, y, z : (x, r_1, z) \wedge (z, r_2, y) \Rightarrow (x, r, y)$ . Here, the relation  $r$  is a composition of  $r_1$  and  $r_2$ ; and the  $\wedge$  is the conjunction operator from relational logic.

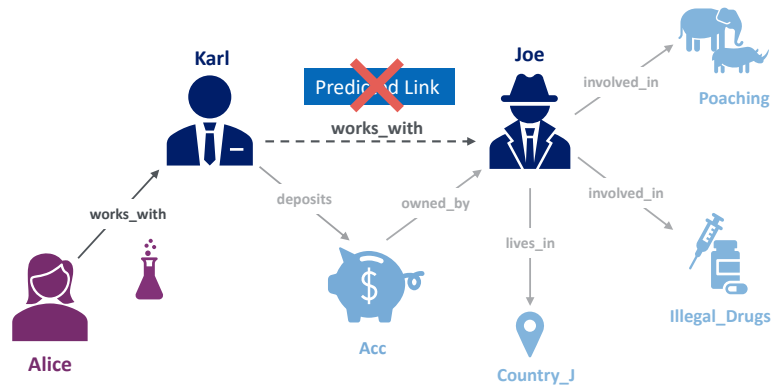
The mapping  $\mathcal{G} : \mathcal{V} \rightarrow \mathcal{E}$  of variables  $\mathcal{V}$  in the above expressions to entities  $\mathcal{E}$  is called a grounding. For example, the logic expression  $\forall x, y : (x, \text{owns}, y) \Rightarrow (y, \text{is\_owned\_by}, x)$  can be mapped to the grounding  $(Karl, \text{owns}, Account) \Rightarrow (Account, \text{is\_owned\_by}, Karl)$ . Thus, a KGE model that captures the inversion pattern will assign a high prediction confidence to the head atom when the body of the clause exists in the knowledge graph.

### 5.3 STEPS FOR ADVERSARIAL ATTACK

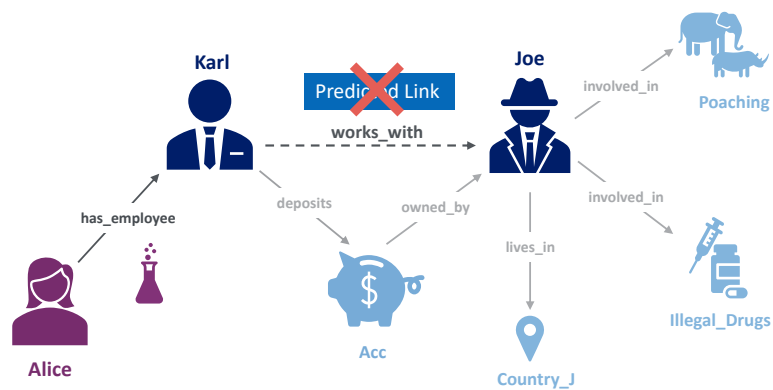
The predictive performance of KGE models is determined by ranking a given fact against the synthetic corruptions of its subject and object entities (Section 2.6.1). Thus, the model's prediction for a target fact can be degraded by carefully selecting a decoy triple from the possible corruptions and adding adversarial triples that improve the rank assigned to the decoy triple. Consider the example of financial knowledge graph from Figure 1 in Chapter 2. Adversarial additions based on the relation inference patterns for this knowledge graph are shown in Figure 7. The target triple whose rank the attacker wants to degrade is  $(Karl, \text{works\_with}, Joe)$ . To degrade the rank for object-side corruptions, the attacker aims to improve the rank of decoy triple  $(Karl, \text{works\_with}, Alice)$ . The decoy triple's rank can be improved through the symmetry pattern by adding  $(Alice, \text{works\_with}, Karl)$ . Similarly, the inversion pattern can be exploited to improve the rank by adding  $(Alice, \text{has\_employee}, Karl)$ . Furthermore, the rank of  $(Karl, \text{works\_with}, Alice)$  can also be improved by exploiting the composition pattern to add the triples  $(Karl, \text{deposits}, Acc\_clean)$  and  $(Acc\_clean, \text{owned\_by}, Alice)$ .

Given this strategy, how should the adversarial attacker select the decoy triple and the corresponding adversarial triples to exploit a specific relation inference pattern? For the logic expressions in the previous section, the decoy triple becomes the head atom and adversarial edits are the triples in the body of the expression. Since the decoy triple is an object or subject side negative of the target triple, the attacker already knows the relation in the head atom. They now want to determine (i) the adversarial relations in the body of the expression; (ii) the decoy entities which will most likely violate the inference pattern for the chosen relations and; (iii) the remaining entities in the body of the expression which will improve the prediction on the chosen decoy triple. Notice that the attacker needs all three steps for composition pattern only; for inversion pattern, only the first two steps

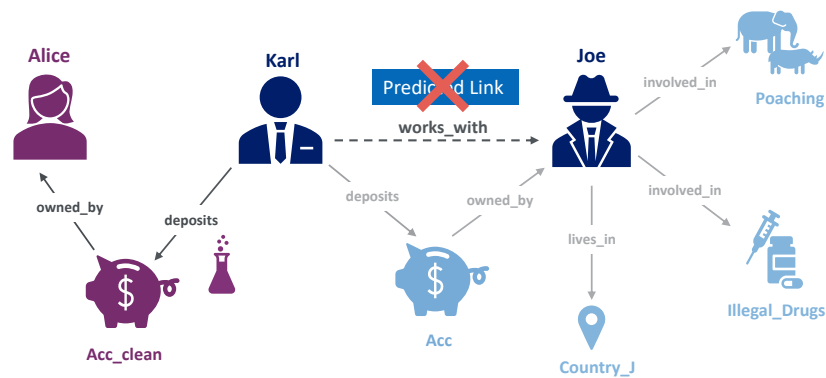
POISONING VIA RELATION INFERENCE PATTERNS



(a) Symmetry Attack



(b) Inversion Attack



(c) Composition Attack

Figure 7: Illustrative examples for adversarial additions based on relation inference patterns. The target triple for attack is  $(Karl, works\_with, Joe)$ , and the decoy triple is  $(Karl, works\_with, Alice)$ . The adversarial addition based on symmetry pattern is  $(Alice, works\_with, Karl)$ , based on inversion pattern  $(Alice, has\_employee, Karl)$ , and based on composition pattern  $(Karl, deposits, Acc\_clean) \wedge (Acc\_clean, owned\_by, Alice)$ .



are needed; and for symmetry pattern, only the second step is needed. Each of these steps is described in detail below.

### 5.3.1 Step1: Determine Adversarial Relations

Expressing the relation patterns as logic expressions is based on relational logic and assumes that the relations are constants. Thus, an algebraic approach is used to determine the relations in the head and body of a clause. Given the target relation  $r$ , the adversarial relations are determined using an algebraic model of inference (Yang et al., 2015). The model is intuitively similar to the prior literature on knowledge graph traversal in latent vector space, used to extract rules from knowledge graphs in Yang et al. (2015), and to answer path queries in knowledge graphs in Guu et al. (2015), Sun et al. (2020a) and Arakelyan et al. (2021).

**INVERSION:** If an atom  $(x, r, y)$  holds true, then for the learned embeddings in multiplicative models, it can be assumed that  $e_x \circ e_r \approx e_y$ ; where  $\circ$  denotes the Hadamard (element-wise) product. If the atom  $(y, r_i, x)$  holds true as well, then it can also be assumed that  $e_y \circ e_{r_i} \approx e_x$ . Thus,  $e_r \circ e_{r_i} \approx \mathbf{1}$  for inverse relations  $r$  and  $r_i$  when the embeddings are learned from multiplicative models. A similar expression is obtained  $e_r + e_{r_i} \approx 0$  when the embeddings are learned from additive models.

Thus, to determine adversarial relations for *inversion* pattern, the pre-trained embeddings are used to select  $r_i$  that minimizes  $|e_{r_i} e_r^T - 1|$  for the multiplicative models; and  $r_i$  that minimizes  $|e_{r_i} + e_r|$  for the additive models.

**COMPOSITION:** If two atoms  $(x, r_1, y)$  and  $(y, r_2, z)$  hold true, then for multiplicative models,  $e_x \circ e_{r_1} \approx e_y$  and  $e_y \circ e_{r_2} \approx e_z$ . Therefore,  $e_x \circ (e_{r_1} \circ e_{r_2}) \approx e_z$ . Hence, relation  $r$  is a composition of  $r_1$  and  $r_2$  if  $e_{r_1} \circ e_{r_2} \approx e_r$ . Similarly, for embeddings from the additive models, the composition pattern is modeled as  $e_{r_1} + e_{r_2} \approx e_r$ .

Thus, to determine adversarial relations for *composition* pattern, the pre-trained embeddings are used to obtain all possible compositions of  $(r_1, r_2)$ . For the multiplicative models, these compositions are obtained using  $e_{r_1} \circ e_{r_2}$  and for additive models, using  $e_{r_1} + e_{r_2}$ . From these compositions, that relation pair is chosen for which the Euclidean distance between the composed relation embeddings and the target relation embedding  $e_r$  is minimum.

### 5.3.2 Step2: Determine Decoy Entities

To select the decoy entity, three different heuristic approaches are considered - soft truth score which is inspired by the research in Guo et al. (2016) and Guo et al. (2018), ranks predicted by the KGE model and cosine distance. The choice of these three approaches is based on the author’s intuition for the most promising ways to select the decoy entities that will violate the relation inference patterns. While other advanced methods might be possible, these simple methods can be implemented readily and efficiently, and serve as good starting points.

**SOFT LOGICAL MODELLING OF INFERENCE PATTERNS:** Once the adversarial relations are determined, it is possible to express the grounding for symmetry, inversion and composition patterns for the decoy triples. Only the object side decoy triple is discussed below for brevity -

$$\begin{aligned}\mathcal{G}_s &: (o', r, s) \Rightarrow (s, r, o') \\ \mathcal{G}_i &: (o', r_1, s) \Rightarrow (s, r, o') \\ \mathcal{G}_c &: (s, r_1, o'') \wedge (o'', r_2, o') \Rightarrow (s, r, o')\end{aligned}$$

If the model captures the patterns  $\mathcal{P}_s$ ,  $\mathcal{P}_i$  or  $\mathcal{P}_c$  to assign high rank to the target triple, then the head atom  $(s, r, o')$  of a grounding that violates this pattern is a suitable decoy triple. Adding the body of this grounding to the knowledge graph would improve the model performance on decoy triple through  $\mathcal{P}_s$ ,  $\mathcal{P}_i$  or  $\mathcal{P}_c$ .

To determine the decoy triple this way, there is need for a measure of the degree to which a grounding satisfies an inference pattern. This measure is called the *soft truth score*  $\phi : \mathcal{G} \rightarrow [0, 1]$  - it provides the truth value of a logic expression indicating the degree to which the expression is true (Guo et al., 2016, 2018). The soft truth score of grounded patterns is modeled using the t-norm based fuzzy logics (Hájek, 1998).

The score  $f_{sro}$  of an individual atom (i.e. triple) is computed using the KGE model’s scoring function. The sigmoid function  $\sigma(x) = 1/(1 + \exp(-x))$  is used to map this score to a continuous truth value in the range  $(0, 1)$ . Hence, the soft truth score for an individual atom is  $\phi(s, r, o) = \sigma(f_{sro})$ . The soft truth score for the grounding of a pattern can then be expressed through logical composition (e.g.  $\wedge$  and  $\Rightarrow$ ) of the scores of individual atoms in the grounding. Following the

research in Guo et al. (2016, 2018), the following compositions are defined for logical conjunction ( $\wedge$ ), disjunction ( $\vee$ ), and negation ( $\neg$ ):

$$\begin{aligned}\phi(a \wedge b) &= \phi(a) \cdot \phi(b), \\ \phi(a \vee b) &= \phi(a) + \phi(b) - \phi(a) \cdot \phi(b), \\ \phi(\neg a) &= 1 - \phi(a).\end{aligned}$$

Here,  $a$  and  $b$  are two logical expressions, which can either be single triples or be constructed by combining triples with logical connectives. If  $a$  is a single triple  $(s, r, o)$ , we have  $\phi(a) = \phi(s, r, o)$ . Given these compositions, the truth value of any logical expression can be calculated recursively (Guo et al., 2016, 2018).

Thus, the following soft truth scores are obtained for the groundings of symmetry, inversion and composition patterns  $\mathcal{G}_s$ ,  $\mathcal{G}_i$  and  $\mathcal{G}_c$  -

$$\begin{aligned}\phi(\mathcal{G}_s) &= \phi(o', r, s) \cdot \phi(s, r, o') - \phi(o', r, s) + 1 \\ \phi(\mathcal{G}_i) &= \phi(o', r_i, s) \cdot \phi(s, r, o') - \phi(o', r_i, s) + 1. \\ \phi(\mathcal{G}_c) &= \phi(s, r_1, o'') \cdot \phi(o'', r_2, o') \cdot \phi(s, r, o') - \phi(s, r_1, o'') \cdot \phi(o'', r_2, o') + 1\end{aligned}$$

To select the decoy triple  $(s, r, o')$  for symmetry and inversion, all possible groundings are scored using  $\phi(\mathcal{G}_s)$  and  $\phi(\mathcal{G}_i)$ . The head atom of the grounding with minimum score is chosen as the decoy triple.

For the composition pattern, the soft truth score  $\phi(\mathcal{G}_c)$  for candidate decoy triples  $(s, r, o')$  contains two entities  $(o', o'')$  to be identified. Thus, a greedy approach is used to select the decoy entity  $o'$ . In this approach, the pre-trained embeddings are used to group the entities  $o''$  into  $k$  clusters using K-means clustering and a decoy entity with minimum soft truth score is determined for each cluster. Then, the decoy entity  $o'$  with minimum score across the  $k$  clusters is selected.

**KGE RANKS:** For this attack heuristic, the ranking protocol from KGE evaluation is used to rank the target triple against the valid subject and object side negatives  $(s', r, o)$  and  $(s, r, o')$ . For each side, that negative triple is selected which is ranked just worse than the target triple (that is,  $negative\_rank = target\_rank + 1$ ). These are suitable as decoy because their predicted scores are likely not very different from the target triple's score. Thus, the model's prediction confidence for these triples might be effectively manipulated through adversarial additions. This is in contrast to very low ranked triples as decoy; where the model has likely learnt a low score with high confidence.

Adversarial Attack Step	Sym	Inv	Com
Determine Adversarial Relations	n/a	Alg	Alg
	Sft	Sft	Sft
Determine Decoy Entities	Rnk	Rnk	Rnk
	Cos	Cos	Cos
Determine Adversarial Entities	n/a	n/a	Sft

Table 15: A summary of the heuristic approaches used for different steps of the adversarial attack with symmetry (Sym), inversion (Inv) and composition (Com) pattern. Alg denotes the algebraic model for inference patterns; Sft denotes the soft truth score; Rnk denotes the KGE ranks; and Cos denotes the cosine distance.

**COSINE DISTANCE:** A high rank for the target triple  $(s, r, o)$  against queries  $(s, r, ?)$  and  $(?, r, o)$  indicates that  $e_s, e_o$  are similar to the embeddings of other subjects and objects related by  $r$  in the training data. Thus, a suitable heuristic for selecting decoy entities  $s'$  and  $o'$  is to choose ones whose embeddings are dissimilar to  $e_s, e_o$ . Since these entities are not likely to occur in the neighbourhood of  $o$  and  $s$ , they will act adversarially to reduce the rank of target triple. Thus, for this attack heuristic, decoy entities  $s'$  and  $o'$  that have maximum cosine distance from target entities  $s$  and  $o$  respectively are selected.

### 5.3.3 Step3: Determine Adversarial Entities

This step is only needed for the composition pattern because the body for this pattern has two adversarial triples. Given the decoy triple in the head of the composition expression, the body of the expression that would maximize the rank of the decoy triple needs to be selected. For this step, the soft-logical model defined in Step 2 for selecting the decoy triples is used again. The soft truth score for composition grounding of the decoy triple is given by  $\phi(\mathcal{G}_t) = \phi(s, r_1, o'') \cdot \phi(o'', r_2, o') \cdot \phi(s, r, o') - \phi(s, r_1, o'') \cdot \phi(o'', r_2, o') + 1$ . The entity  $o''$  with maximum score is selected as the adversarial entity. This is because this entity satisfies the composition pattern for the decoy triple and is thus likely to improve the decoy triple's ranks on addition to the knowledge graph.

## 5.4 COMPUTATIONAL COMPLEXITY ANALYSIS

The main challenge in designing adversarial additions against KGE models is the large combinatorial search space of the candidate triples. To overcome this chal-

lenge, the attacks based on relation inference patterns break down the search space into three smaller steps. The computational complexity of the proposed attacks is discussed below. Let's say  $\mathcal{E}$  is the set of entities and  $\mathcal{R}$  is the set of relations. The number of target triples to attack is  $t$  and the specific target triple is  $(s, r, o)$ .

**DETERMINE ADVERSARIAL RELATIONS:** This step determines the inverse relation or the composition relation of a target triple. To select the inverse relation,  $\mathcal{R}$  computations are needed for every target triple. On the other hand, selecting composition relation requires the composition operation  $\mathcal{R}^2$  times per target triple. To avoid the repetition of these computations for every target triple, the inverse and composition relations are pre-computed for all relations in the knowledge graph. This gives the complexity  $\mathcal{O}(\mathcal{R}^2)$  for the inverse relation. For the composition relation, compositions of all relation pairs are computed and the adversarial pair is selected by comparison with the target relation. This gives  $\mathcal{O}(\mathcal{R}^2 + \mathcal{R})$  complexity for composition. The computational complexity for this step is independent of the number of target triples  $t$ .

**DETERMINE DECOY ENTITY:** For a given target triple, the three heuristics to select the decoy entity are soft-truth score, KGE ranks and cosine distance. The expression for computing the soft-truth scores requires the scores for individual atoms in the expression for all possible entity substitutions. Using a pre-trained KGE model, one atomic score value for all possible entity substitutions can be computed by one forward call to the model. This way, for symmetry and inversion, computing the soft truth score values requires 2 forward calls to the KGE model because these expressions contain 2 different atomic expressions. For the composition pattern, the expression for soft truth scores contains three different atomic expressions. Furthermore, since the expression contains an entity  $o''$  in addition to the decoy entity, the score needs to be computed for  $k$  different clusters into which the entities are grouped. Thus, computing the soft truth values of the composition pattern for one target triple requires  $3k$  forward calls to the model.

Selection of the decoy entity based on KGE ranks requires the computation of one atomic expression for all possible entity substitutions. This requires one forward call to the pre-trained KGE model. The atomic scores further need to be ranked using Pytorch's `torch.sort` function. In general, the sorting operation has a time complexity of  $\mathcal{O}(\mathcal{E} \log(\mathcal{E}))$ . Thus, the expected computational complexity for selecting the decoy entity for one target triple using the ranks from the KGE model is  $\mathcal{O}(\mathcal{E} \log(\mathcal{E}))$ .

For cosine distance, the similarity of  $s$  and  $o$  to all entities is computed via two calls to Pytorch’s `F.cosine_similarity`. Once the heuristic scores are computed, there is an additional complexity of  $\mathcal{O}(\mathcal{E})$  to select the entity with minimum score. These computations need to be repeated for all target triples  $t$ . Thus, the complexity for decoy selection is  $\mathcal{O}(t\mathcal{E})$  for all heuristics except two cases - soft truth score on composition where it is  $\mathcal{O}(kt\mathcal{E})$  and scores based on KGE ranks where it is  $\mathcal{O}(t(\mathcal{E}\log(\mathcal{E}) + \mathcal{E}))$ , which can be simplified to  $\mathcal{O}(t\mathcal{E}\log(\mathcal{E}))$ .

**DETERMINE ADVERSARIAL ENTITY:** This step requires the computation of ground truth scores for the composition pattern which contain three different atomic expressions. Thus, the step requires three forward calls to the KGE model. The computational complexity of this step for all target triples is  $\mathcal{O}(t\mathcal{E})$ .

Based on the above discussion, the computational complexity for different attacks is as follows -

1. For symmetry attacks based on soft-truth scores and cosine similarity, the complexity is  $\mathcal{O}(t\mathcal{E})$ ; and for the symmetry attacks based on KGE ranks, the complexity is  $\mathcal{O}(t\mathcal{E}\log(\mathcal{E}))$
2. For inversion attacks based on soft-truth scores and cosine similarity, the complexity is  $\mathcal{O}(\mathcal{R}^2 + t\mathcal{E})$ ; and for the inversion attacks based on KGE ranks, the complexity is  $\mathcal{O}(\mathcal{R}^2 + t\mathcal{E}\log(\mathcal{E}))$
3. For composition attacks, the complexity is  $\mathcal{O}(\mathcal{R}^2 + \mathcal{R} + kt\mathcal{E})$  for soft truth score. On simplification, this complexity expression becomes  $\mathcal{O}(\mathcal{R}^2 + kt\mathcal{E})$ . The complexity for KGE ranks is  $\mathcal{O}(\mathcal{R}^2 + \mathcal{R} + t\mathcal{E}\log(\mathcal{E}))$ , which on simplification becomes  $\mathcal{O}(\mathcal{R}^2 + t\mathcal{E}\log(\mathcal{E}))$ . For composition attacks based on cosine distance, the time complexity is  $\mathcal{O}(\mathcal{R}^2 + \mathcal{R} + t\mathcal{E})$ , which can be simplified to  $\mathcal{O}(\mathcal{R}^2 + t\mathcal{E})$ .

It is noteworthy that the above expressions for computational complexity indicate that all the proposed attack strategies are more efficient than the naive method of searching through the combinatorial space of all possible adversarial additions. For  $t$  target triples, the computational complexity for the enumerative search would be  $\mathcal{O}(t\mathcal{R}\mathcal{E})$ . This expression does not account for the complexity of computing the metric for the impact of a candidate on the target triple. Despite this, the computational complexity of the complete attack strategies discussed above is more efficient than the naive methods. By breaking down the search space into smaller steps, the proposed attack strategies reduce the multiplicative components of the search space complexity into additive components.

## 5.5 EXPERIMENTAL SETUP

As discussed in Section 3.6, the aim of the evaluation is to assess the effectiveness of proposed attacks in *degrading* the predictive performance of KGE models on missing triples that are predicted true. For this, the state-of-the-art evaluation protocol for data poisoning attacks (Xu et al., 2020) has been used. First, a clean model is trained on the original data; then the adversarial edits are generated and added to the training knowledge graph; and finally a new model is re-trained on this poisoned graph. All the hyperparameters for training the KGE model on original and poisoned datasets remain the same.

### 5.5.1 Datasets

The proposed attack strategies are evaluated on four models with varying inductive abilities - DistMult, ComplEx, ConvE and TransE; on two publicly available benchmark datasets for link prediction<sup>1</sup>- WN18RR and FB15k-237. To evaluate the predictive performance of the KGE models, standard KGE evaluation protocol is used (Section 2.6.1). In addition, following Bordes et al. (2013) and as is the accepted state-of-the-art practice, triples from the validation and test set that contain unseen entities are filtered out for the KGE model evaluation.

To assess the attack effectiveness in *degrading* performance on triples predicted as True, a set of triples that are predicted as True by the model are needed. Thus, a subset of the original test set is selected as *target triples*, where each triple is ranked  $\leq 10$  by the original model. That is, the subset of test set that is ranked the best by the original model is selected as target triples for the evaluation. Table 16 provides an overview of dataset statistics and the number of target triples selected.

### 5.5.2 Baselines

The proposed attack strategies are compared against the following baselines based on random edits as well as state-of-art data poisoning attacks.

*Random<sub>n</sub>*: Random edits in the neighbourhood of each entity of the target triple.

*Random<sub>g1</sub>*: Global random edits in the knowledge graph which are not restricted to the neighbourhood of entities in the target triple and have 1 edit per decoy triple (like symmetry and inversion).

<sup>1</sup> <https://github.com/TimDettmers/ConvE>

	WN18RR	FB15k-237	
Entities	40,559	14,505	
Relations	11	237	
Training	86,835	272,115	
Validation	2,824	17,526	
Test	2,924	20,438	
	DistMult	1,315	3,342
Target	ComplEx	1,369	3,930
	ConvE	1,247	4,711
	TransE	1,195	5,359

Table 16: Statistics for the datasets WN18RR and FB15k-237. Triples from the validation and test set that contained unseen entities were removed to ensure that new entities are not added as adversarial edits to the training graph. The numbers above (including the number of entities) reflect this filtering.

*Random\_g2*: Global random edits in the knowledge graph which are not restricted to the neighbourhood of entities in the target triple and have 2 edits per decoy triple (like composition).

*Direct-Add*: Poisoning attack from Zhang et al. (2019a) for edits in the neighbourhood of subject of the target triple. The method is extended to both subject and object side neighbours to match the evaluation protocol in this chapter. Further implementation details of this attack are available in Appendix B.1.2.

*CRIAGE*: Poisoning attack from Pezeshkpour et al. (2019). The publicly available implementation of the attack and the default attack settings<sup>2</sup> are used for this research. The method was proposed for edits in the neighbourhood of object of the target triple. It is extended to the neighbourhood of both subject and object entities to match the evaluation protocol in this research and to ensure fair evaluation.

### 5.5.3 Implementation

For every attack, the adversarial edit candidates that already exist in the graph are filtered out. This post-processing step is done to ensure that the poisoned training graph does not contain duplicate triples. Similarly, duplicate adversarial edits for different targets are also removed before adding them to the original knowledge graph. For Step 2 of the composition attack with ground truth, the elbow method is used to determine the number of clusters for each model-data combination. The

<sup>2</sup> <https://github.com/pouyapez/criage>



	DistMult		Complex		ConvE		TransE	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
<b>Original</b>	0.90	0.85	0.89	0.84	0.92	0.89	0.36	0.03
<b>Baseline Attacks</b>								
Random_n	0.86 (-4%)	0.83	0.84 (-6%)	0.80	0.90 (-2%)	0.88	0.28 (-20%)	0.01
Random_g1	0.88	0.83	0.88	0.83	0.92	0.89	0.35	0.02
Random_g2	0.88	0.83	0.88	0.83	0.91	0.89	0.34	0.02
Direct-Add	0.82 (-8%)	0.81	0.76 (-14%)	0.74	0.90 (-2%)	0.87	<b>0.24 (-33%)</b>	<b>0.01</b>
CRIAGE	0.87	0.84	-	-	0.90	0.88	-	-
<b>Proposed Attacks</b>								
Sym_truth	0.66	0.40	<b>0.56 (-37%)</b>	<b>0.24</b>	<b>0.61 (-34%)</b>	<b>0.28</b>	0.57	0.36
Sym_rank	0.61	0.32	<b>0.56 (-37%)</b>	<b>0.24</b>	0.62	0.31	0.25	0.02
Sym_cos	<b>0.57 (-36%)</b>	<b>0.32</b>	0.62	0.43	0.67	0.44	<b>0.24 (-33%)</b>	<b>0.01</b>
Inv_truth	0.87	0.83	0.86	0.80	0.90	0.87	0.34	0.03
Inv_rank	0.86	0.83	0.85	0.80	0.89 (-4%)	0.85	0.25	0.02
Inv_cos	0.83 (-8%)	0.82	0.80 (-10%)	0.79	0.90	0.88	0.25 (-30%)	0.01
Com_truth	0.86	0.83	0.86	0.81	0.89	0.86	0.53 (+49%)	0.27
Com_rank	0.85 (-5%)	0.80	0.83	0.77	0.89	0.84	0.57	0.32
Com_cos	0.86	0.77	0.82 (-8%)	0.70	0.88(-4%)	0.83	0.53 (+49%)	0.27

Table 17: Reduction in MRR and Hits@1 due to different attacks on the **target split** of WN18RR. For each block of rows, the *best* relative percentage difference from original MRR; computed as  $(original - poisoned) / original * 100$  is reported. Lower values indicate better results; best results for each model are in bold. Statistics on the target split are in Table 16.

elbow method is a heuristic approach to select the optimal number of clusters in the k-means clustering, that is, the value of k.

Further details on KGE model training, computing resources and number of clusters used are available in Appendix B.1. The source code to reproduce the experimental results reported in this Chapter is available on GitHub at <https://github.com/PeruBhardwaj/InferenceAttack>.

## 5.6 EVALUATION OF ATTACK EFFECTIVENESS

Table 17 and 18 show the reduction in MRR and Hits@1 due to different attacks on the WN18RR and FB15k-237 datasets. These results show that the proposed adversarial attacks outperform the random baselines and the state-of-art data poisoning attacks against all KGE models on both datasets.

	DistMult		Complex		ConvE		TransE	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
<b>Original</b>	0.61	0.38	0.61	0.45	0.61	0.45	0.63	0.48
<b>Baseline Attacks</b>								
Random_n	0.54 (-11%)	0.40	0.54 (-12%)	0.40	0.56 (-8%)	0.41	0.60 (-4%)	0.45
Random_g1	0.54	0.40	0.55	0.41	0.57	0.43	0.62	0.46
Random_g2	0.55	0.41	0.55	0.40	0.57	0.42	0.61	0.46
Direct-Add	0.53 (-13%)	0.39	0.51 (-16%)	0.38	0.54 (-11%)	0.39	0.57 (-10%)	0.42
CRIAGE	0.54	0.41	-	-	0.56	0.41	-	-
<b>Proposed Attacks</b>								
Sym_truth	0.51	0.36	0.56	0.41	<b>0.51 (-17%)</b>	<b>0.34</b>	0.62	0.48
Sym_rank	0.53	0.39	0.53	<b>0.38</b>	0.55	0.38	<b>0.53 (-16%)</b>	<b>0.36</b>
Sym_cos	<b>0.46 (-25%)</b>	<b>0.31</b>	<b>0.51 (-17%)</b>	<b>0.38</b>	0.52	0.37	0.55	0.40
Inv_truth	0.55	0.41	0.54	0.40	0.56	0.41	0.62	0.46
Inv_rank	0.56	0.43	0.55	0.40	0.55 (-9%)	0.40	0.58 (-8%)	0.42
Inv_cos	0.54 (-11%)	0.40	0.53 (-14%)	0.39	0.56	0.42	0.59	0.44
Com_truth	0.56	0.42	0.55	0.41	0.57	0.43	0.65	0.51
Com_rank	0.56 (-8%)	0.42	0.55 (-11%)	0.40	0.56 (-8%)	0.41	0.69	0.48
Com_cos	0.56 (-8%)	0.43	0.56	0.42	0.56	0.42	0.63 (0%)	0.49

Table 18: Reduction in MRR and Hits@1 due to different attacks on the **target split** of FB15k-237. For each block of rows, the *best* relative percentage difference from original MRR; computed as  $(original - poisoned) / original * 100$  is reported. Lower values indicate better results; best results for each model are in bold. Statistics on the target split are in Table 16.

### 5.6.1 Comparison across Relation Inference Patterns

It is observed that the attacks based on symmetry inference pattern perform the best across all model-dataset combinations. This indicates the sensitivity of KGE models to symmetry pattern. For DistMult, ComplEx and ConvE, this sensitivity can be explained by the symmetric nature of the scoring functions of these models. That is, the models assign either equal or similar scores to triples that are symmetric opposite of each other. In the case of TransE, the model’s sensitivity to symmetry pattern is explained by the translation operation in scoring function. The score of target  $(s, r, o)$  is a translation from subject to object embedding through the relation embedding. Symmetry attack adds the adversarial triple  $(o', r, s)$  where the relation is same as the target relation, but target subject is the object of adversarial triple. Now, the model learns the embedding of  $s$  as a translation from  $o'$  through relation  $r$ . This adversarially modifies the embedding of  $s$  and in turn, the score of  $(s, r, o)$ .

Furthermore, the inversion and composition based attacks perform better than the baselines in most cases, but not as good as symmetry. For the composition pattern, it is likely that the model has stronger bias for shorter and simpler patterns like symmetry and inversion than for composition. This makes it harder to deceive the model through composition than through symmetry or inverse.

### 5.6.2 Comparison of Datasets

The above observation for the inversion and composition pattern is particularly true for FB15k-237 where the performance for these patterns is similar to random baselines. One possible reason for the low effectiveness on FB15k-237 dataset could be the high connectivity of this dataset (Dettmers et al., 2018). This means that the KGE model relies on a high number of triples to learn the target triples’ ranks. Thus, effectively poisoning the KGE models for FB15k-237 will likely require more adversarial triples per target triple than that considered in this research.

The inversion pattern is likely ineffective on the benchmark datasets because these datasets do not have any inverse relations (Dettmers et al., 2018; Toutanova and Chen, 2015). This implies that the attacks cannot identify the inverse of the target triple’s relation in Step 1 and a KGE model trained on these clean datasets would not be vulnerable to inversion attacks. To investigate this hypothesis further, the attacks are evaluated on WN18 dataset where the inverse relations have not been filtered out. This means that the KGE model can learn the inversion pattern and the inversion attacks can identify the inverse of the target relation.

	DistMult		Complex		ConvE		TransE	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
<b>Original</b>	0.82	0.67	0.99	0.99	0.80	0.63	0.65	0.45
<b>Baseline Attacks</b>								
Random_n	0.80 (-2%)	0.63	0.99 (0%)	0.98	0.79 (-2%)	0.61	0.46 (-29%)	0.18
Random_g1	0.82	0.66	0.99	0.98	0.80	0.62	0.57	0.33
Random_g2	0.81	0.65	0.99	0.98	0.79	0.62	0.50	0.22
Direct-Add	0.77 (-6%)	0.59	0.97 (-3%)	0.95	0.77 (-3%)	0.61	0.43 (-33%)	0.16
CRIAGE	0.78	0.61	-	-	0.78	0.63	-	-
<b>Proposed Attacks</b>								
Sym_truth	0.62	0.30	0.90	0.82	<b>0.58 (-17%)</b>	<b>0.27</b>	0.74	0.60
Sym_rank	0.59	0.27	0.89 (-10%)	0.79	0.62	0.33	0.52	0.34
Sym_cos	<b>0.50 (-38%)</b>	<b>0.17</b>	0.92	0.85	0.60	0.35	<b>0.41 (-37%)</b>	<b>0.13</b>
Inv_truth	0.81	0.66	0.86	0.74	0.78 (-3%)	0.61	0.59	0.34
Inv_rank	0.82	0.66	<b>0.84 (-16%)</b>	<b>0.68</b>	0.79	0.61	0.55	0.34
Inv_cos	0.79 (-3%)	0.64	0.87	0.75	0.80	0.63	0.51 (-22%)	0.25
Com_truth	0.79	0.62	0.98	0.97	0.77	0.62	0.53 (-18%)	0.25
Com_rank	0.80	0.64	0.98	0.96	0.75 (-6%)	0.58	0.67	0.47
Com_cos	0.78 (-5%)	0.61	0.97 (-2%)	0.95	0.77	0.62	0.58	0.32

Table 19: Reduction in MRR and Hits@1 due to different attacks on the **target split** of WN18. For each block of rows, the *best* relative percentage difference from original MRR; computed as  $(original - poisoned) / original * 100$  are reported. Lower values indicate better results; best results for each model are in bold.

Table 19 shows the results for different adversarial attacks on WN18. In this setting, it is observed that the inversion attacks outperform other attacks against Complex on WN18, indicating the sensitivity of Complex to the inversion pattern when the dataset contains inverse relations. It is also observed that the symmetry based attack is the most effective for DistMult, ConvE and TransE. This indicates the sensitivity of these models to the symmetry pattern even when the inverse relations are present in the knowledge graph. For DistMult and ConvE, this is likely due to the symmetric nature of their scoring functions; and for TransE, this is likely because of the translation operation as discussed earlier.

### 5.6.3 Analysis on Decoy Triples

An exception in the results is the composition pattern on TransE where the model performance improves instead of degrading on the target triples. This is likely due to the model’s sensitivity to composition pattern such that adding this pattern

improves the performance on all triples, including target triples. To verify this, the change in ranks of the decoy triples was investigated.

The proposed attacks are designed to generate adversarial triples that improve the KGE model performance on decoy triples  $(s, r, o')$  and  $(s', r, o)$ . Thus, an analysis was done to determine whether the performance of KGE models improves or degrades over decoy triples after poisoning. For the decoy triples on object side  $(s, r, o')$ , the change in object side MRR was computed relative to the original object side MRR of these triples. Similarly, for the decoy triples on subject side  $(s', r, o)$ , the change in subject side MRR was computed relative to the original subject side MRR of these decoy triples. Figure 8 shows the plots for the mean change in MRR of object and subject side decoy triples.

It can be noticed that the composition attacks against TransE are effective in improving the ranks of decoy triples on both WN18RR and FB15k-237. Thus, it is likely that the composition attack does not work against TransE for WN18RR because the original dataset does not contain any composition relations; thus adding this pattern improves model's performance on *all* triples instead of just the target triples because of the sensitivity of TransE to composition pattern.

It also explains why the increase is more significant for WN18RR than FB15k-237. WN18RR does not have any composition relations but FB15k-237 does; thus, adding these to WN18RR shows significant improvement in performance. This behaviour for the composition pattern also indicates that the selection of adversarial entities in Step 3 of the composition attacks can be improved.

Further investigation of these and additional hypotheses about the proposed adversarial attacks are interesting directions for future work.

## 5.7 ADDITIONAL EVALUATION

### 5.7.1 *Analysis of Runtime Efficiency*

In this section, the runtime efficiency of the baseline and proposed attacks is compared. Given the original KGE model and the set of target triples, the absolute time taken to generate the complete set of adversarial additions for all target triples is recorded. Table 20 shows the time taken (in seconds) to select the adversarial triples using different attack strategies for all models on WN18 dataset. Similar patterns were observed for attack execution on other datasets.

For CRIAGE, the reported time does not include the time taken to train the auto-encoder model for that attack. Similarly, for soft-truth based composition attacks, the reported time does not include the time taken to pre-compute the

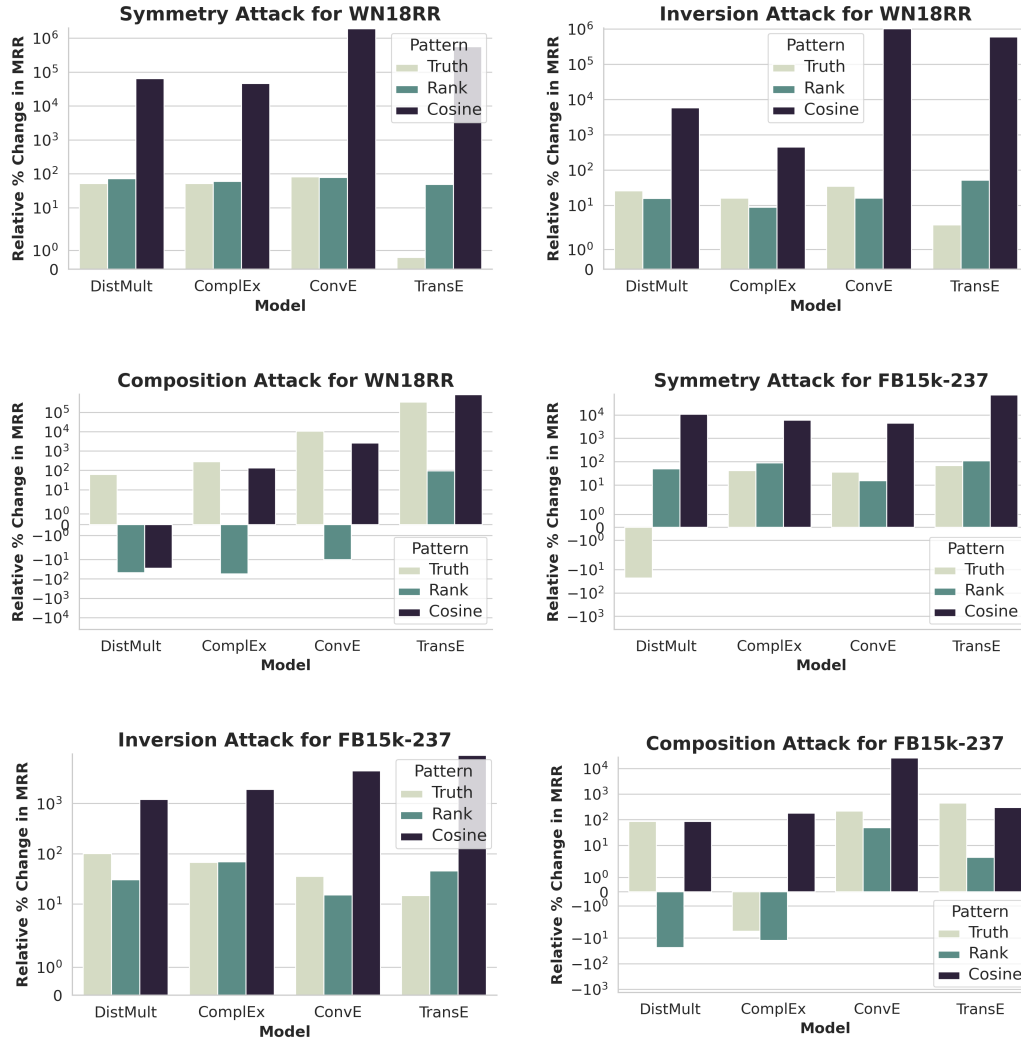


Figure 8: Mean of the relative increase in MRR of object and subject side **decoy triples** due to proposed attacks on WN18RR and FB15k-237. The increase is computed relative to the original MRR of decoy triples as  $(poisoned - original)/original$ . The scale on the y-axis is a symmetric log scale. Higher values are better; as they show the effectiveness of attack in improving decoy triples' ranks relative to their original ranks.

		DistMult	ComplEx	ConvE	TransE
<b>Baseline Attacks</b>	Random_n	10.08	10.69	8.76	7.83
	Random_g1	<b>8.28</b>	<b>8.16</b>	<b>7.64</b>	<b>6.49</b>
	Random_g2	16.01	15.82	18.72	13.33
	Direct-Add	94.48	<b>255.53</b>	666.85	<b>81.96</b>
	CRIAGE	<b>21.77</b>	-	<b>21.96</b>	-
<b>Proposed Attacks</b>	Sym_truth	<b>19.63</b>	35.40	<b>22.76</b>	31.59
	Sym_rank	23.47	<b>27.25</b>	25.82	25.03
	Sym_cos	22.52	28.62	25.69	<b>23.13</b>
	Inv_truth	<b>11.43</b>	<b>15.69</b>	24.13	31.89
	Inv_rank	15.27	18.14	30.99	21.82
	Inv_cos	14.96	20.47	<b>23.02</b>	<b>20.63</b>
	Com_truth	2749.60	1574.44	6069.79	470.34
	Com_rank	<b>22.04</b>	<b>31.53</b>	37.81	20.88
	Com_cos	34.78	68.06	<b>32.37</b>	<b>19.86</b>

Table 20: Runtime efficiency of the baseline and proposed adversarial attacks against different KGE models for WN18 dataset. The absolute time taken in seconds to generate the complete set of adversarial additions for all the target triples is reported.

clusters of entities. It is observed that the proposed attacks are more efficient than the baseline Direct-Add attack which requires a combinatorial search over the candidate adversarial triples. In addition, the proposed attacks have comparable efficiency to CRIAGE. Among the different proposed attacks, composition attacks based on soft-truth score take more time than others because they select the decoy entity by computing the soft-truth score for multiple clusters.

### 5.7.2 Qualitative Analysis

Since adversarial attacks cause model failure for specific predictions, these failure points can be used to understand the predictive performance of KGE models. This section provides examples of the adversarial additions generated using the Inference Attacks. Tables 21 and 22 show the adversarial additions for target triples which undergo the maximum change in ranks due to poisoning. While symmetry and inversion patterns have two adversarial additions, the composition pattern has four. Table 21 provides examples for WN18RR ComplEx and Table 22 provides examples for FB15k-237 ComplEx. The entities in the benchmark data splits are represented as numeric IDs not string expressions. The string representations of entities for Wordnet (WN18RR) are obtained using the *definitions.txt* file from the original split at <https://everest.hds.utc.fr/doku.php?id=en:smemlj12>. For

Freebase (FB15k-237), the entity representations are obtained using a community build browser at <https://freebase.toolforge.org/>. The values for maximum change in ranks for WN18RR are  $-(13385.5, 34866, 40554)$  for Symmetry attacks,  $(22212.5, 40292.5, 40549.5)$  for Inversion attacks, and  $(17150.5, 40460, 35417)$  for Composition attacks. On the other hand, these values for FB15k-237 are  $-(1292, 596, 4400.5)$  for Symmetry,  $(832, 1119, 1903)$  for Inversion, and  $(675, 614, 2180.5)$  for Composition.

As indicated by the results in Tables 17 and 18, the maximum change in ranks due to poisoning of WN18RR is significantly more than FB15k-237. This observation was explained above in terms of the graph connectivity - since FB15k-237 has more neighbours for the target triple, the KGE model prediction relies on a larger number of triples and is thus, more difficult to perturb than WN18RR. Consider the examples for the Sym\_truth attacks. For WN18RR, the target (*date\_NN\_6*, *hypernym*, *month\_NN\_1*) has 11 neighbouring triples, whereas for FB15k-237, the triple (*Oprah Winfrey*, *award/ ranking/ list*, *Time 100*) has 76 neighbours. Furthermore, the triple with minimum change in ranks for the Sym\_truth attack on WN18RR is (*family\_bruchidae\_NN\_1*, *hypernym*, *arthropod\_family\_NN\_1*). Though this triple has the same target relation *hypernym*, the number of neighbouring triples are 146. These values provide further support that the KGE model’s predictions for triples with fewer neighbours are more vulnerable to adversarial attacks than those for triples with more neighbours.

For both datasets, it can be observed that different attack methods cause maximum change to the ranks of different target triples, though the same set of target triples is used for all attacks. This is likely because the predictions for different target triples are sensitive to different inference patterns or attack strategies. However, the relation *hypernym* is common among the target triples for WN18RR.

Furthermore, the adversarial relations for symmetry pattern are not necessarily symmetric. For example, for the symmetry attack on WN18RR, the relations *hypernym* and *derivationally\_related\_form* are most sensitive. However, only *derivationally\_related\_form* is an actual symmetric relation because it relates similar words with different syntactic categories or parts-of-speech. The relation *hypernym* denotes supertype or generalization, and is not symmetric. Similarly, for FB15k-237, */people/ person/ religion* is not a symmetric relation. These adversarial relations are, however, still effective likely due to the training graph structure, which makes the model prediction sensitive to this inference pattern. Consider the target triple (*date\_NN\_6*, *hypernym*, *month\_NN\_1*) from WN18RR. The entity *date\_NN\_6* is related to *date\_VB\_3* for 2 triples, and has the relation *hypernym* for a third triple. Similarly, the entity *month\_NN\_1* is related via *hypernym* to 7 entities, and via *has\_part*



to *week\_NN\_3*. However, there is no direct or 2-hop link between the target entities. Thus, connecting (*date\_VB\_3* , *date\_NN\_6*) and (*month\_NN\_1* , *week\_NN\_3*) via the *hypernym* relation improves the ranks of decoy triples and reduces the likelihood of assigning a high rank to the target triple.

Among the target triples for inversion pattern, for FB15k-237, the model identifies */location/hud\_county\_place/place* as the inverse of */people/ethnicity/people*. While the two relations are not exact inverses of each other, their domain and range would loosely be inverse. Among the target triples for composition pattern, the model identifies *film/performance/film* and *location/ mailing\_address/ citytown* as the composition for */people/person/gender*.

### 5.7.3 Comparison with Attribution Attacks

In Chapter 4, the instance attribution methods were used to select adversarial additions by identifying the most influential triple for a target triple, and replacing one of the entities of the target triple with a dissimilar entity in the latent space. While this heuristic method overcomes the search space for adversarial additions, it still requires a search over the neighbourhood of the target triple. This section compares the effectiveness and efficiency of the attribution attacks from Chapter 4 with the inference attacks proposed in this Chapter. The experimental setup is the same as the remaining experiments in this chapter - that is, two adversarial additions are crafted per target triple. To select two adversarial additions from attribution attacks, instead of the most influential triple, the top-2 most influential triples are selected and their entities are replaced with dissimilar entities.

Table 23 shows the predictive performance of ComplEx for WN18RR and FB15k-237 due to the attribution attacks and inference attacks. It is observed that the symmetry pattern based attacks outperform the other methods for both datasets. Furthermore, the instance attribution attacks perform similar to the inversion and composition patterns. As discussed in previous sections, this indicates the sensitivity of ComplEx to the symmetry pattern. However, another explanation for the better performance of inference attacks is that they exploit the KGE ranking protocol for evaluation. According to the KGE evaluation protocol (Section 2.6.1), the existence of a missing target fact is decided by its rank against the possible entity corruptions. While attribution attacks target the score predicted for the target facts, inference attacks target their predicted rank. Thus, the inference attack strategy is likely more effective than attribution attacks. This also indicates the possibility of designing stronger adversarial attacks against KGE models by jointly targeting the predicted score as well as the rank of the target fact.

Attack	Target and Adversarial Triples
<b>Sym</b>	<i>date_NN_6 , hypernym , month_NN_1</i>
<b>Truth</b>	<i>date_VB_3 , hypernym , date_NN_6 month_NN_1 , hypernym , week_NN_3</i>
<b>Sym</b>	<i>roman_deity_NN_1 , hypernym , immortal_NN_2</i>
<b>Rank</b>	<i>greco-roman_deity_NN_1 , hypernym , roman_deity_NN_1 immortal_NN_2 , hypernym , mythology_NN_2</i>
<b>Sym</b>	<i>scratch_VB_3 , derivationally_related_form , scabies_NN_1</i>
<b>Cos</b>	<i>itch_VB_2 , derivationally_related_form , scratch_VB_3 scabies_NN_1 , derivationally_related_form , itching_NN_1</i>
<b>Inv</b>	<i>pyrrhocoridae_NN_1 , hypernym , arthropod_family_NN_1</i>
<b>Truth</b>	<i>suborder_heteroptera_NN_1 , derivationally_related_form , pyrrhocoridae_NN_1 arthropod_family_NN_1 , derivationally_related_form , kingdom_animalia_NN_1</i>
<b>Inv</b>	<i>roman_deity_NN_1 , hypernym , immortal_NN_2</i>
<b>Rank</b>	<i>greco-roman_deity_NN_1 , derivationally_related_form , roman_deity_NN_1 immortal_NN_2 , derivationally_related_form , mythology_NN_2</i>
<b>Inv</b>	<i>flying_lizard_NN_1 , hypernym , agamid_lizard_NN_1</i>
<b>Cos</b>	<i>family_agamidae_NN_1 , derivationally_related_form , flying_lizard_NN_1 agamid_lizard_NN_1 , derivationally_related_form , genus_draco_NN_1</i>
<b>Com</b>	<i>republic_of_guinea_NN_1 , instance_hyponym , african_nation_NN_1</i>
<b>Truth</b>	<i>republic_of_guinea_NN_1 , hypernym , konakri_NN_1 konakri_NN_1 , member_of_domain_usage , wetback_NN_1  terminate_VB_1 , hypernym , republic_of_cameroon_NN_1 republic_of_cameroon_NN_1 , member_of_domain_usage , african_nation_NN_1</i>
<b>Com</b>	<i>zaglossus_NN_1 , member_meronym , spiny_anteater_NN_1</i>
<b>Rank</b>	<i>zaglossus_NN_1 , derivationally_related_form , tachyglossidae_NN_1 tachyglossidae_NN_1 , hypernym , pipistrellus_pipistrellus_NN_1  zaglossus_NN_1 , derivationally_related_form , tachyglossidae_NN_1 tachyglossidae_NN_1 , hypernym , spiny_anteater_NN_1</i>
<b>Com</b>	<i>garnish_NN_2 , hypernym , ornamentation_NN_2</i>
<b>Cos</b>	<i>garnish_NN_2 , derivationally_related_form , interior_decoration_NN_1 interior_decoration_NN_1 , member_meronym , dress_VB_9  trim_VB_6 , derivationally_related_form , grace_VB_2 grace_VB_2 , member_meronym , ornamentation_NN_2</i>

Table 21: Examples of target triples from WN18RR ComplEx with maximum change in rank due to adversarial attacks. For each target triple, the adversarial additions are shown for degrading ranks on the object-side (*s,r,?*) and the subject-side (*? ,r,o*) queries. Symmetry and inversion attacks have two additions per target triple, whereas composition attacks have four additions per target triple.

Attack	Target and Adversarial Triples		
<b>Sym Truth</b>	Oprah Winfrey	/award/ ranked_item/ appears_in_ranked_lists./ award/ ranking/ list	Time 100
	The Women of Brewster Place	/award/ ranked_item/ appears_in_ranked_lists./ award/ ranking/ list	Oprah Winfrey
	Time 100	/award/ ranked_item/ appears_in_ranked_lists./ award/ ranking/ list	Sarah Palin
<b>Sym Rank</b>	Stanley Kubrick	/people/ person/ religion	Judaism
	Agnosticism	/people/ person/ religion	Stanley Kubrick
	Judaism	/people/ person/ religion	Sam Mendes
<b>Sym Cos</b>	Lieutenant	/business/ job_title/ people_with_this_title./ business/ employment_tenure/ company	United States Navy
	United States of America	/business/ job_title/ people_with_this_title./ business/ employment_tenure/ company	Lieutenant
	United States Navy	/business/ job_title/ people_with_this_title./ business/ employment_tenure/ company	United States Army
<b>Inv Truth</b>	Italian American	/people/ ethnicity/ people	John Travolta
	American English	/location/ hud_county_place/ place	Italian American
	John Travolta	/location/ hud_county_place/ place	United States of America
<b>Inv Rank</b>	Coldplay	/common/ topic/ webpage./ common/ webpage/ category	Official Website
	Lead vocalist	/tv/ tv_program/ languages	Coldplay
	Official Website	/tv/ tv_program/ languages	Jay-Z
<b>Inv Cos</b>	The Royal Conservatoire of Scotland	/education/ educational_institution/ students_graduates./ education/ education/ student	Alan Cumming
	Tony Award for Best Actor in a Musical	/film/ film/ featured_film_locations	The Royal Conservatoire of Scotland
	Alan Cumming	/film/ film/ featured_film_locations	United States of America

(a) Symmetry and Inversion Attacks

Table 22: Examples of target triples from FB15k-237 ComplEx with maximum change in rank due to adversarial attacks. For each target triple, the adversarial additions are shown for degrading ranks on the object-side ( $s, r, ?$ ) and the subject-side ( $? , r, o$ ) queries. Symmetry and inversion attacks have two additions per target triple, whereas composition attacks have four additions per target triple.

<b>Attack</b>	<b>Target and Adversarial Triples</b>		
<b>Com Truth</b>	Bryan Adams	/people/ person/ gender	Male
	Bryan Adams	/film/ actor/ film./ film/ performance/ film	St. John's University
	St. John's University	/organization/ organization/ headquarters./ location/ mailing_address/ citytown	New York City
	Jane Lynch Autism	/film/ actor/ film./ film/ performance/ film /organization/ organization/ headquarters./ location/ mailing_address/ citytown	Autism Male
<b>Com Rank</b>	Mila Kunis	/people/ person/ gender	Female
	Mila Kunis Autism	/film/ actor/ film./ film/ performance/ film /organization/ organization/ headquarters./ location/ mailing_address/ citytown	Autism Male
	Lily Tomlin Into the Wild	/film/ actor/ film./ film/ performance/ film /organization/ organization/ headquarters./ location/ mailing_address/ citytown	Into the Wild Female
<b>Com Cos</b>	Pink Floyd	/common/ topic/ webpage./ common/ webpage/ category	Official Website
	Pink Floyd	/award/ award_winning_work/ awards_won./ award/ award_honor/ award_winner	Zoey Deschanel
	Zoey Deschanel	/people/ person/ spouse_s./ people/ marriage/ location_of_ceremony	Los Angeles
	Grammy Award for Best Rock Vocal Performance by a Duo or Group	/award/ award_winning_work/ awards_won./ award/ award_honor/ award_winner	Eagles
	Eagles	/people/ person/ spouse_s./ people/ marriage/ location_of_ceremony	Official Website

(b) Composition Attacks

Table 22: Examples of target triples from FB15k-237 ComplEx (continued).

		WN18RR		FB15k-237	
		MRR	H@1	MRR	H@1
<b>Original</b>		0.89	0.84	0.61	0.45
<b>Attribution Attacks</b>	Dot Metric	0.85(-5%)	0.80	0.55(-10%)	0.42
	$\ell_2$ Metric	0.80(-10%)	0.75	0.54(-12%)	0.40
	Cos Metric	0.80(-10%)	0.74	0.54(-12%)	0.41
	GD (dot)	0.76(-15%)	0.65	0.54(-12%)	0.40
	GL ( $\ell_2$ )	0.75(-16%)	0.65	0.54(-13%)	0.40
	GC (cos)	0.75(-16%)	0.65	0.54(-11%)	0.41
<b>Inference Attacks</b>	Sym	<b>0.56(-37%)</b>	<b>0.24</b>	<b>0.51(-17%)</b>	<b>0.38</b>
	Inv	0.80(-10%)	0.79	0.53(-14%)	0.39
	Com	0.82(-8%)	0.70	0.55(-11%)	0.40

Table 23: Comparison of the effectiveness of adversarial additions based on Attribution Attacks and Inference Attacks for the ComplEx model on WN18RR and FB15k-237. For Inference Attacks, the best results for each relation inference pattern from Tables 17 and 18 is reported here. Percentage values were computed before rounding off to two decimals.

Table 24 provides a comparison of the runtime efficiency of different attack methods against DistMult model for WN18RR and FB15k-237. It is observed that the symmetry attacks are most efficient for both datasets - performing similar to the random baselines for WN18RR and better than the random baselines for FB15k-237. Additionally, as observed for adversarial deletions in Section 4.6.1, the attribution attacks (especially gradient similarity based attribution) are more efficient for WN18RR than for FB15k-237. This is because of their reliance on the number of triples in the neighbourhood of the target triple. On the other hand, the difference in the efficiency of inference attacks for WN18RR and FB15k-237 is similar to the difference in the efficiency of random baselines. This is because the efficiency of inference attacks depends on the number of relations, entities and target triples instead of neighbourhood triples. This observation provides empirical evidence to support the computational complexity analysis of inference patterns based attacks from Section 5.4 of this Chapter.

		WN18RR	FB15k-237
<b>Baseline Attacks</b>	Random_n	2.57	<b>25.35</b>
	Random_g1	<b>2.13</b>	25.38
	Random_g2	2.46	31.25
	Direct-Add	19.30	257.68
	CRIAGE	<b>4.91</b>	<b>6.03</b>
<b>Attribution Attacks</b>	Dot Metric	7.57	13.34
	$\ell_2$ Metric	<b>7.43</b>	<b>9.94</b>
	Cos Metric	7.56	11.47
	GD (dot)	<b>39.53</b>	<b>1810.68</b>
	GL ( $\ell_2$ )	44.09	1885.22
	GC (cos)	52.47	2130.87
<b>Inference Attacks</b>	Sym_truth	<b>3.02</b>	10.53
	Sym_rank	3.08	10.33
	Sym_cos	3.06	<b>9.72</b>
	Inv_truth	5.84	10.66
	Inv_rank	5.77	10.48
	Inv_cos	<b>5.65</b>	<b>10.05</b>
	Com_truth	988.02	915.64
	Com_rank	8.73	23.37
	Com_cos	<b>8.02</b>	<b>22.05</b>

Table 24: Runtime efficiency of the baseline and proposed adversarial additions for DistMult on WN18RR and FB15k-237. The absolute time taken in seconds to generate the complete set of adversarial triples for all target triples is reported. For adversarial additions based on Attribution attacks, neighbourhood triples for each target triple were pre-computed.

## 5.8 SUMMARY

This chapter proposed data poisoning attacks against KGE models based on relation inference patterns like symmetry, inversion and composition. The evaluation of these attacks showed that the proposed attacks outperform state-of-the-art adversarial attacks. Since the proposed attacks rely on relation inference patterns, they can also be used to understand the predictive behaviour of KGE models. This is because if a KGE model is sensitive to a relation inference pattern, then that pattern should be an effective adversarial attack. It was observed that the attacks based on symmetry pattern generalize across all KGE models. This indicates the sensitivity of different KGE models to this inference pattern.

Further investigation of the hypotheses about the effect of input graph connectivity and the existence of specific inference patterns in the datasets are interesting directions for future work. It is noteworthy that such investigation of attacks will likely be influenced by the choice of datasets. In this thesis, the benchmark datasets for link prediction (Section 2.6.3) have been used for evaluation. While there are intuitive assumptions about the inference patterns on these datasets, there is no study that formally measures and characterizes the existence of these inference patterns. This makes it challenging to verify the claims made about the inductive abilities of the KGE models, not only by the attacks proposed in this chapter but also by new KGE models proposed in the literature.

Thus, a promising step in understanding knowledge graph embeddings is to propose datasets and evaluation tasks that test varying degrees of specific inductive abilities (Section 6.2.3). These will help evaluate new models and serve as a testbed for poisoning attacks. Furthermore, specifications of model performance on datasets with different inference patterns will improve the usability of KGE models in high-stakes domains like healthcare and finance.

---

## RELATED WORK

---

The research in this thesis focuses on the adversarial vulnerabilities of representation learning for knowledge graphs. The aim of this chapter is to relate this focus to the wider literature on graph representation learning and on understanding the predictive behaviour of latent knowledge graph representations. The discussion in this chapter is not necessary to understand the main contributions of the thesis. However, these studies might serve as useful pointers for an interested reader.

### 6.1 GRAPH REPRESENTATION LEARNING

Traditional methods for Machine Learning on graphs relied on hand engineered features which are extracted from the structure of the graph (Hamilton et al., 2017b; Hamilton, 2020). More recent approaches instead learn these features through representation learning approaches. A graph representation learning algorithm encodes the structural information in a graph by learning a mapping from the nodes (or sub-graphs) into low dimensional feature vectors. The objective is to optimize the mapping such that embedding interactions in lower dimensional vector space reflect the structure of the original graph. A large number of approaches have been proposed for representation learning on graphs. Hamilton et al. (2017b); Hamilton (2020) proposed a unifying framework for the different representation learning approaches on graphs. Broadly, there are three families of algorithms based on the graph structural information they aim to preserve – edges between entities, neighbourhood nodes of an entity, relational paths between entities.

#### 6.1.1 *Based on Edge Reconstruction*

Representation learning algorithms which have an objective function based on edge reconstruction, as proposed in Cai et al. (2018), are designed particularly for relational graphs or knowledge graphs where the edges in the graph represent re-



lations between entities. The entire relational graph can be considered a set of facts about the entities, represented as subject-relation-object triples. The most common ML task on such directed, labelled graphs is that of knowledge base completion by predicting the missing facts (also called link prediction) (Nickel et al., 2011; Bordes et al., 2013; Yang et al., 2015). The learned embeddings, popularly called Knowledge Graph Embeddings (KGE), and the task of knowledge base completion are the focus of the research in this thesis, and are discussed in Chapter 2.

The embedding algorithms rely on local and global patterns of relationships to generalize them. Node embeddings are learned by low-rank tensor factorization of the tensor representing the graph. Tensor factorization is an extension of matrix factorization where two matrices are used to represent the set of subjects and objects. However, since the same entity can occur as subject and object, Nickel et al. (2011) proposed RESCAL to learn joint embeddings for subjects and objects. A matrix of scores for the occurrence of a fact (or triple) can be obtained by a multilinear product between the embeddings of subject, relation and object. This notion of dot product has been generalized to a ranking based scoring function which combines embedding vectors in such a way that the actual triples in the graph are ranked higher than those which are not in the graph (Cai et al., 2018; Trouillon et al., 2016). Representation learning techniques based on ranking score based edge reconstruction differ mostly in the way they define this embedding combination or scoring function. As proposed in Chandrhas et al. (2018), TransE (Bordes et al., 2013) and TransR (Lin et al., 2015) use additive scoring functions while DistMult (Yang et al., 2015), HoLE (Nickel et al., 2016b) and ComplEx (Trouillon et al., 2016) use multiplicative scoring functions. In TransE (Bordes et al., 2013), relations are interpreted as translations on low-dimensional entity embeddings and the object entity embedding is approximated as embedding of subject entity plus relation embedding. On the other hand, DistMult (Yang et al., 2015) computes the score matrix as a dot product of the subject embedding, object embedding and relation matrix. This is extended in ComplEx (Trouillon et al., 2016) by replacing the dot product with a Hermitian product for complex values. Further discussion on the design of KGE models used for the research in this thesis, as well as the training and evaluation protocols for this family of models is available in Chapter 2.

The KGE methods discussed above focus on simple knowledge graphs where the entities belong to a fixed, discrete set, and the graph is represented as RDF (Resource Description Framework) triples. However, KG representation can also include multimodal entities, relation attributes or a graph schema definition in Web Ontology Language (OWL). Pezeshkpour et al. (2018); García-Durán and Niepert (2018); Kristiadi et al. (2019); Pai and Costabello (2021) propose differ-

ent approaches to encode the multimodal information like images, text, numerical values and logical rules, and combine this encoding with latent vectors from simple KGE models. [Lacroix et al. \(2020\)](#); [Messner et al. \(2022\)](#) extend the ComplEx and BoxE models to temporal KGs with timestamped relations. [Chen et al. \(2021\)](#) proposes embeddings for OWL ontologies that represent the domain semantics through logic constructors and lexical information. The research in this thesis focuses on the KGE models for simple graph structures expressed as RDF triples.

### 6.1.2 *Based on Neighbourhood Aggregation*

Neighbourhood aggregation based representation learning relies on a node's local neighbourhood to learn its embedding. This is unlike edge reconstruction based techniques that utilize the local and global relation patterns in the graph. In neighbourhood aggregation, node attributes are used to initialize node embeddings. This initial embedding is iteratively updated using an aggregation function on the vectors representing neighbourhood nodes. Over the iterations, information from further nodes is aggregated into this node embedding ([Hamilton et al., 2017b](#)).

A neural network based approach for such neighbourhood aggregation, called Graph Neural Network (GNN) was first proposed in [Scarselli et al. \(2009\)](#). Inspired by traditional Recurrent Neural Network (RNN), this algorithm relied on repeated application of contraction maps as propagation (or aggregation) function until stable node embeddings were achieved. The GNN architecture was extended further in [Li et al. \(2016\)](#) by using Gated Recurrent Unit (GRU). Instead of this RNN based aggregation, neighbourhood aggregation based on convolution operations was proposed in [Bruna et al. \(2014\)](#) and extended in [Defferrard et al. \(2016\)](#). The traditional Convolutional Neural Network (CNN) has been successful on Euclidean data which can be represented by a grid-like structure or vectors. These studies extend the convolution operation to graphs (non-Euclidean data) by using Graph Laplacian in the spectral domain. The spectral graph convolution operation is further simplified in [Kipf and Welling \(2017\)](#) via a localized first order approximation which is equivalent to an element wise weighted mean aggregation function (spatial convolution). The use of convolution operations for aggregation is generalized to trainable aggregation functions in [Hamilton et al. \(2017a\)](#). [Bronstein et al. \(2017\)](#) provides an overview of additional similar methods.

Most graph representation learning algorithms based on neighbourhood aggregation are designed for undirected, unlabelled graphs. But, some of these have been generalized to relational (directed, labelled) graphs for link prediction. For example, R-GCN ([Schlichtkrull et al., 2018](#)) is a generalization of Graph Convo-

lutional Network (GCN) (Kipf and Welling, 2017) for relational graphs. However, the architecture does not scale to practical-scale knowledge graphs. This is because GNNs aggregate the neighbourhood information for each node, implying that the same neighbours need to be sampled multiple times. This quickly becomes a bottleneck for GNN implementation. For smaller graphs, the bottleneck is resolved by performing the convolution operation on the complete graph adjacency matrix at once (Hamilton, 2020). However, this solution cannot be applied for large adjacency tensors for practical knowledge graphs. Therefore, neighbourhood aggregation based representation learning algorithms are not included in this thesis.

### 6.1.3 *Based on Random Walks*

Random walks based representation learning on graphs attempts to preserve the first order proximity or the second order proximity of a node in low dimensional vectors. First proximity refers to local pairwise proximity between nodes, while second order proximity refers to the proximity between the neighbourhood structures of nodes (Tang et al., 2015). Node embeddings are generated by sampling a set of paths from the graph and maximizing the probability of observing a node's neighbourhood (Cai et al., 2018). Hence, nodes with similar neighbourhood have similar embeddings. The techniques differ in their sampling strategy and the node proximity to be preserved.

DeepWalk (Perozzi et al., 2014) uses truncated random walks to sample the paths in graph. These paths are treated like sequences of words where the nodes correspond to words. A SkipGram model is used to learn the embeddings by optimizing a hierarchical softmax function. LINE (Tang et al., 2015) uses a Breadth-first sampling (BFS) strategy to expand a node's context while sampling. The BFS approach captures homophily, where nodes belonging to same clusters are embedded closely. On the other hand, a Depth-first Sampling (DFS) strategy captures structural equivalence, where nodes with similar structural roles in the graph are embedded closely. Both BFS and DFS are used for expansion of a node's context in node2vec (Grover and Leskovec, 2016) via hyperparameters to bias the random walks. Both LINE and node2vec use negative sampling for optimizing the objective function. While DeepWalk preserves a node's second order proximity, LINE preserves both the first and second order proximity. node2vec preserves the first order proximity for homophilic and structurally equivalent nodes.

Since these methods encode the neighbourhood structural information to latent representations through random walks, they do not scale to knowledge graphs (Hamilton, 2020). Hence, these methods are not included for research in this thesis.

## 6.2 UNDERSTANDING KNOWLEDGE GRAPH REPRESENTATIONS

The latent distributed representations of knowledge graphs have achieved impressive predictive performance on a variety of knowledge-driven downstream tasks like question answering, fact checking, relation extraction, entity disambiguation and so on (Nickel et al., 2016a; Ji et al., 2022; Chen et al., 2020). This has made representation learning algorithms the most promising methods for reasoning with knowledge graphs in real-world scenarios. The graph representation learning algorithms are indeed increasingly used in high-stakes domains like healthcare and finance (Rotmensch et al., 2017; Mohamed et al., 2020; Bonner et al., 2022).

However, deploying ML systems in real-world scenarios brings new challenges about the reliability and trustworthiness of these systems. The learned models are difficult to interpret and thus require novel methods to explain their predictions. Being black-box, the failure modes of the models are also unknown, which motivates research for designing adversarial attacks and investigating the security vulnerabilities of the models. Additionally, the reasons for impressive predictive performance of the black-box models are unknown. This performance is often achieved by extensive tuning of the model hyperparameters, which motivates research for dissecting and understanding the impact of these hyperparameters on the model predictions.

While the research in this thesis focuses on the adversarial vulnerabilities of Knowledge Graph Embeddings, complementary research on understanding the embeddings includes methods for post-hoc explanations, improved evaluation protocols, theoretical and empirical analyses, etc. This section highlights some literature on these related topics. A brief recap of state-of-the-art methods for adversarial attacks is included for completeness, but a deeper analysis relevant for the thesis contributions is available in Section 3.4.

### 6.2.1 *Via Robustness of Model Predictions*

Zhang et al. (2019a) and Pezeshkpour et al. (2019) propose data poisoning attacks against KGE models and are the most closely related to this research. These attacks are used as baselines to evaluate the thesis contributions. Additionally, Lawrence et al. (2021) estimates the influence of training triples on the KGE model predictions to provide post-hoc explanations. The proposed method can also be used for adversarial deletions and is thus, included as a baseline in this research.

In parallel work, Banerjee et al. (2021) studies risk aware adversarial attacks against KGE models with the aim of reducing the exposure risk of an adversarial

attack instead of improving the attack effectiveness. This study is not included in the evaluation protocol for the thesis as the research here focuses on improving the attack effectiveness. [Zhang et al. \(2021\)](#) investigates the adversarial vulnerabilities of cross-lingual knowledge alignment through perturbations of knowledge graphs. Another recent study [Raman et al. \(2021\)](#) designs deceptive perturbations to the knowledge graph that maximally modify the graph structure while preserving the performance of downstream tasks on the graph. The study aims to investigate the role of knowledge graphs in knowledge-driven applications. Most recently, [Betz et al. \(2022\)](#) proposes adversarial attacks against KGE models for gray-box settings. Similar to the adversarial attacks in Chapter 5 of the thesis, relation inference patterns are exploited to select adversarial edits. However, instead of extracting the inferred patterns from the learned embedding, the study mines logical rules directly from the knowledge graph. This enables the selection of adversarial deletions and additions without knowledge of the victim KGE model.

[Pujara et al. \(2017\)](#) also investigates the failure modes of the KGE models. However, instead of designing adversarial perturbations to the input knowledge graph, the study generates systematic variations of the datasets. Using this suite of datasets, the sensitivity of KGE models to sparse and unreliable data is examined.

Besides data poisoning attacks, [Minervini et al. \(2017\)](#) uses adversarial regularization in the latent space to inject the background knowledge expressed as First Order Logic (FOL) clauses into KGE models. Embedding perturbations that do not comply with the logical constraints are penalized through an inconsistency loss. Given useful prior constraints, the regularization approach improves the predictive performance of KGE models for link prediction. However, these adversarial samples are not in the input domain and aim to improve instead of degrade the model performance. Similarly, [Cai and Wang \(2018\)](#) uses adversarial learning to improve the negative sampling process during the KGE model training.

In addition to the risk of adversarial manipulation of training data, trustworthy deployment of KGE models also needs to address unintentional changes in the data distribution. In particular, [Fisher et al. \(2020\)](#) shows that KGE models can encode social biases represented in the training data, and result in unfair predictions for certain populations. The study further proposes a debiasing method to train KGE models that are neutral towards sensitive attributes. This research is extended in [Keidar et al. \(2021\)](#) by identifying the sensitive relations to be debiased automatically. On the other hand, [Du et al. \(2022\)](#) proposes novel measures to quantify the gender bias in KG embeddings and attributes the origin of this bias to training instances using Influence Function ([Koh and Liang, 2017](#)).

### 6.2.2 *Via Explainability of Model Predictions*

Given a trained KGE model, a post-hoc explanation method explains the model predictions on specific target facts. The post-hoc explanations can constitute triples, relation paths, or sub-graphs from the input knowledge graph. Lawrence et al. (2021) proposes Gradient Rollback to estimate the influence of training triples on model predictions and provides a set of top-k most influential triples as post-hoc explanation. Zhang et al. (2019c) provides explanations for KGE model predictions as meaningful paths between the subject and object of the triple to be explained. The study uses the embeddings of entities and relations to search for reliable explanation paths. In addition to closed paths between the subject and object, similar explanation structures from the remaining graph are also generated as support for the main explanation. Nandwani et al. (2020) also provides relation paths as explanations but focuses on tensor factorization based KGE models. Similarity between the entity embeddings is used to augment the knowledge graph with weighted edges and similar explanation paths are aggregated into second-order templates. A separate neural module is trained to select the best template as the explanation.

Evaluating the methods for post-hoc KGE model explanations can be challenging because of the lack of ground-truth explanations. Thus, Halliwell et al. (2021) created a benchmark dataset of explanations for missing link prediction on knowledge graphs. This dataset is built using a semantic reasoner on a fixed set of rules.

Unlike the non-interpretable nature of latent representations, rule-based symbolic reasoning on knowledge graphs is inherently interpretable (Chen et al., 2020). This has led to research efforts for combining rule-based reasoning with black-box latent representations to improve the latter’s explainability. One research direction here constraints the knowledge graph embedding space by using background logical facts. Guo et al. (2016) and Guo et al. (2018) enforce soft logical rules by modelling the triples and rules in a unified framework and jointly learning embeddings from them. Additional studies that inject logical rules into knowledge graph embeddings include Rocktäschel et al. (2015), Demeester et al. (2016) and Minervini et al. (2017). Further, García-Durán and Niepert (2018) combines the logical rules with latent and numerical features in an end-to-end framework using the product of experts approach. The soft-logical model of relation inference patterns, proposed for adversarial attacks in Chapter 5 of this thesis is inspired by the literature on joint modelling of facts and rules (Guo et al., 2016, 2018).

While rule-based reasoning is interpretable, logical rules are often not readily available. The methods described above rely on state-of-the-art rule mining systems like AMIE+ (Galárraga et al., 2015) and AnyBURL (Meilicke et al., 2019) to

obtain the background logical facts. However, mining arbitrary rules from knowledge bases can be computationally expensive and constraining the type of rules to be extracted requires a manual specification prior to rule extraction (Chen et al., 2020). Further, injecting the logical rules to neural networks requires groundings of the rules, the size of which can be very large (Cohen et al., 2020). Thus, another research direction in combining symbolic rules with black-box neural embeddings focuses on the joint learning of logical rules and latent features from knowledge graphs. Neural Theorem Provers (NTPs) (Rocktäschel and Riedel, 2017) replace symbols in the logical rules with differentiable vector representations, which allows the rules to be learned from the knowledge graphs through backward chaining. However, the search space for NTPs is exponentially large. This issue is addressed in Minervini et al. (2020a,b) by selecting the rules greedily or conditionally. In Arakelyan et al. (2021), the NTPs are further extended to answer complex path queries on knowledge graphs in an efficient and explainable manner. On the other hand, Qu et al. (2021) uses a probabilistic approach for rule learning that treats logic rules as latent variables. The use of a rule generator and reasoning predictor effectively reduces the search space. Another system for learning rules from knowledge graphs is MINERVA (Das et al., 2018), which uses a reinforcement learning agent to learn a policy for graph traversal as an optimal sequence of decisions. Further, Das et al. (2020) proposes a non-parametric approach based on case-based reasoning that does not rely on latent embeddings for reasoning. Similarly, end-to-end differentiable systems like NeuraLP (Yang et al., 2017) and DRUM (Sadeghian et al., 2019) learn logical rules without latent features.

Despite these advances in interpretable methods for knowledge base reasoning, the rule injection methods remain difficult to scale and the rule learning methods only perform well in simple settings (de Jong and Sha, 2019). Thus, black-box KGE models remain the most popular method for reasoning on knowledge graphs.

### 6.2.3 *Via Miscellaneous Methods*

Over the years, a large number of KGE models have been proposed, and have achieved impressive predictive performance. However, due to multiple components in the KGE training pipeline, there is wide variation in the implementation, training and evaluation of the proposed models (Ali et al., 2021). Thus, dissecting and investigating the impact of different training components has become important for understanding the KGE model behaviour. One of the earlier studies in this direction was Kadlec et al. (2017) which showed that simple KGE architectures can outperform newer and more sophisticated ones by proper hyperparameter tuning.

Following this, [Mohamed et al. \(2019\)](#) discovered that the KGE model's predictive performance is not only impacted by its scoring function, but also by the loss function used to train the model. Furthermore, [Lacroix et al. \(2018\)](#) showed that the model performance can be improved for the same scoring function by adding reciprocal triples to the training dataset.

In response to these findings, studies on re-evaluating and benchmarking the predictive performance of KGE models have emerged. [Ruffinelli et al. \(2020\)](#) evaluated five KGE models with different loss functions and training strategies. The study found that early scoring functions like RESCAL ([Nickel et al., 2011](#)) show comparable performance to more recent and advanced architectures when trained with state-of-the-art techniques. More recently, this study has been extended in [Ali et al. \(2021\)](#) to perform a comprehensive benchmarking of the KGE model performance. Another recent survey [Rossi et al. \(2021\)](#) investigates the effect of different graph structural properties on the predictive performance of KGE models.

Besides understanding the training phase of KGE models, some studies have focused on identifying inconsistencies in the evaluation protocol. [Sun et al. \(2020b\)](#) highlighted that the strong predictive performance of KGE models can be attributed to an incorrect evaluation implementation. The study further proposed an evaluation protocol for assigning the ranks when multiple triples have the same score. In the same direction, [Pezeshkpour et al. \(2020\)](#) highlighted the unreliability of the standard evaluation metrics and [Rim et al. \(2021\)](#) proposed to evaluate KGE models using checklists ([Ribeiro et al., 2020](#)).

In the literature besides benchmarking and reproducibility, [Trouillon et al. \(2019\)](#) studies the inductive abilities of KGE models as binary relation properties for controlled inference tasks with synthetic datasets. On the theoretical side, [Wang et al. \(2018\)](#) studies the expressiveness of various bilinear KGE models and [Gutiérrez-Basulto and Schockaert \(2018\)](#) studies the ability of KGE models to learn hard rules expressed as ontological knowledge. More recently, [Allen et al. \(2021\)](#) provides a theoretical connection between the word embeddings and knowledge graph representations. The study proposes a model for encoding the semantic graph relationships as geometric interactions between the latent representations.

### 6.3 SUMMARY

This chapter provided a discussion of the related work for this thesis. The first part discussed families of graph representation learning algorithms including those that do not scale to knowledge graphs. The second part highlighted complementary literature on understanding the predictive behaviour of KGE models.



---

## CONCLUSION

---

### 7.1 CONTRIBUTIONS

This thesis presents the research undertaken towards the adversarial robustness of representation learning on knowledge graphs through the design, implementation and evaluation of data poisoning attacks against them. Representation learning on knowledge graphs is increasingly used in high-stakes domains where the decisions based on the learned representations impact human lives. However, by virtue of being data driven, the representation learning algorithms are black-box and thus, have unknown failure modes. Responsible deployment of representation learning algorithms for knowledge graphs in user-facing applications requires an investigation of their security vulnerabilities and robustness against adversarial attacks. Towards this investigation, the thesis contributes two data poisoning attacks that *effectively* degrade the predictive performance of the Knowledge Graph Embeddings at inference time by removing or adding triples to the knowledge graph at training time.

1. The first contribution of the thesis is poisoning attacks based on instance attribution methods (Chapter 4). The instance attribution methods from Interpretable Machine Learning identify the input data points that are most influential to a model's prediction. These methods are used to select the influential triples as adversarial deletions, that is triples in the training graph that on removal degrade the learned model's predictive performance. Furthermore, a heuristic approach is proposed to select adversarial additions based on the influential triples.
2. The second contribution of the thesis is poisoning attacks based on the relation inference patterns (Chapter 5). These attacks exploit the inductive abilities of KGE models which are expressed as connectivity patterns among the relationships in the knowledge graph, for example, symmetry, inversion

## CONCLUSION

and composition. To degrade the KGE model’s predicted ranks on target triples, different relation inference patterns are exploited to improve the performance on decoy triples. This problem formulation breaks down the original combinatorial search space for adversarial additions into three smaller steps for determining the adversarial relations, the decoy entities that most likely violate an inference pattern and the remaining adversarial entities in the inference pattern that are most likely to improve the rank of decoy triples. Unlike the adversarial additions based on instance attribution methods, executing these attacks does not require an enumeration over the neighbourhood triples of the target triple. Instead, the computational complexity is an additive function of the number of entities and relations.

The proposed attacks are evaluated by comparing the predictive performance of the original and the poisoned KGE models on the target triples that are ranked best by the original model. The experiments for four KGE models DistMult, ComplEx, ConvE and TransE on two benchmark datasets WN18RR and FB15k-237 indicate that the proposed attacks are more effective than the baselines based on random edits and state-of-the-art adversarial attacks against KGE models. It was found that the attacks based on simpler instance attribution methods and simpler relation patterns are more or as effective as the more complex attacks. Thus, in a hypothetical white-box attack scenario, an attacker might efficiently execute highly effective adversarial attacks against Knowledge Graph Embeddings. Additionally, models from all the different representative families are observed to be vulnerable to degradation in their predictive performance. The proposed attacks were also found to be more effective for the sparser WN18RR dataset than the FB15k-237. This indicates that the embeddings learned from fewer training triples are more vulnerable to adversarial manipulation of the training knowledge graph.

The KGE models evaluated in this research have shown competitive predictive performance in recent literature (Kadlec et al., 2017; Ruffinelli et al., 2020) through proper hyperparameter tuning. This makes them promising candidates for use in production pipelines. However, the research in this thesis highlights the brittleness of these performance gains and thus, calls for improved KGE model evaluation that accounts for adversarial robustness in addition to the predictive performance.

## 7.2 LIMITATIONS

While this thesis clearly highlights the adversarial vulnerabilities of KGE models, there are some constraints and limitations of the research presented in the thesis.

### 7.2.1 Evaluation Design

The conclusions drawn from this research are based on evaluation of the proposed attacks against four KGE models on two benchmark datasets. The choices made towards KGE model selection, dataset selection and implementation design have likely influenced the outcomes of the evaluation. For example, a large number of KGE models have been proposed in the literature, but this research focuses on four representative models from the three families of KGE models (tensor decomposition, geometric and deep learning models as discussed in Rossi et al. (2021)). While the KGE models are differentiated through their scoring function design, each KGE model can be trained through multiple loss functions like pointwise, pairwise or setwise loss functions (Ali et al., 2021). Additionally, there are multiple training strategies like negative sampling, 1vsAll and KvsAll (Ruffinelli et al., 2020). It is very likely that the effectiveness of the proposed adversarial attacks would vary for different scoring functions, loss functions and training design.

In the recent literature, multiple studies (Ali et al., 2021; Rossi et al., 2021; Kadlec et al., 2017; Ruffinelli et al., 2020) on benchmarking the predictive performance of KGE models have been conducted. While these studies focus on the reproducibility of the predictive performance, their design and implementation suggests that a similar study for evaluating the adversarial robustness of KGE models is beyond the scope of this thesis. Nevertheless, a detailed comparative study of the adversarial vulnerabilities of KGE models is a useful direction for future work.

In addition to above design choices, the conclusions rely on characteristics of the datasets chosen in this research. The datasets used here are the benchmark datasets for link prediction using KGE models. It was observed that the instance attribution attacks are more effective on the sparser dataset WN18RR. Similarly, the addition of inversion inference pattern to WN18RR affected the effectiveness of relation inference patterns attacks against ComplEx. However, the characteristics of these benchmark datasets need not necessarily be representative of the real-world knowledge graphs in safety-critical domains. For example, the qualitative analysis in Section 4.6.2 showed that identifying the correct influential triples for FB15k-237 is challenging because of the semantically incoherent triples in the test set. Additionally, there are limited, though only recent, studies (Cao et al., 2021; Ali et al., 2021) that formally measure and characterize the connectivity patterns or relation inference patterns in benchmark datasets for link prediction. This makes it challenging to generalize the claims made about the effectiveness of adversarial attacks in this research. Analyzing the impact of dataset characteristics on the effectiveness of adversarial attacks is an interesting direction for future work.

### 7.2.2 Design of Adversarial Attacks

The relation inference patterns based attacks in this thesis use three steps, of which, the first step is to determine the adversarial relations (Table 15). This step utilizes the relation embeddings from KGE models to select the inverse or the composition relations for the given target triple. The algebraic model of inference patterns used for this selection assumes that the scoring function of KGE model is either additive or multiplicative in nature (Chandrasah et al., 2018). However, some recent KGE models, like RotatE (Sun et al., 2019) and MuRE (Balazevic et al., 2019a), have both or neither additive and multiplicative components in their scoring functions. Hence, the proposed attacks based on relation inference patterns are only applicable for KGE models with additive or multiplicative scoring functions.

## 7.3 BROADER IMPACT

This thesis studies the problem of generating data poisoning attacks against KGE models. Data poisoning attacks identify the vulnerabilities in learning algorithms that could be exploited by an adversary to manipulate the model’s behaviour (Joseph et al., 2019; Biggio and Roli, 2018). Such manipulation can lead to unintended model behaviour and failure. Identifying these vulnerabilities for KGE models is critical because of their increasing use in domains that need high stakes decision making like healthcare (Rotmensch et al., 2017; Mohamed et al., 2020) and finance (Hogan et al., 2021; Noy et al., 2019). In this way, the research presented here aims to safeguard the KGE models against potential harm from adversaries and thus, minimize the negative consequences of deploying these models in user facing applications that require trustworthy predictions.

Arguably, because the vulnerabilities are studied by attacking the KGE models, the proposed attacks can be used by an actual adversary to manipulate the predictive behaviour of models in deployed systems. This paradox of an arms race is universal across security research (Biggio and Roli, 2018). For this thesis, the principle of proactive security design, as recommended by Joseph et al. (2019) and Biggio and Roli (2018) has been followed. As opposed to reactive security measures where the learning system designers develop countermeasures after the system is attacked, a proactive approach *anticipates* such attacks, simulates them and designs countermeasures before the systems are deployed. Therefore, by revealing the vulnerabilities of KGE, this thesis provides an opportunity to fix them.

Besides the use case of security evaluation, the research can be helpful in understanding the predictive behaviour of KGE models. To propose adversarial attacks

based on adversarial deletions, the research uses Instance Attribution methods from Interpretable Machine Learning. These methods identify the most influential triples in the training knowledge graph which can also be used as post-hoc explanations for the learned KGE models (Section 4.6.2). In addition to explaining model predictions, instance based attribution methods can help guide the design decisions during KGE model training. As discussed earlier, there are a vast number of KGE model architectures, loss functions and training strategies. Empirically quantifying the impact of different design choices on the predictive performance of KGE models is often challenging and requires extensive experimental resources (Ruffinelli et al., 2020; Ali et al., 2021; Rossi et al., 2021). Thus, it would be very promising to explore the use of instance attribution methods to understand the impact of these choices on the KGE model predictions. By tracing back the model predictions to the input knowledge graph, one can gain a better understanding of the success or failure of different design choices.

The proposed attacks based on relation inference patterns can also be used to understand the inductive abilities of KGE models. These attacks rely on the inductive assumptions of a model, which are expressed as the different relation inference patterns, to be able to deceive that model. Thus, theoretically, the effectiveness of attacks based on one inference pattern over another indicates the model's reliance on one inference pattern over another. However, as discussed in the previous section, it is challenging to generalize the conclusions drawn from this research about the inductive abilities of KGE models. This is because the inference patterns for benchmark datasets are not well defined, and these datasets may not be representative of the high-stakes domains.

Thus, there is scope for further research to evaluate the adversarial attacks proposed in this thesis. A promising direction here is to design benchmark tasks and datasets that measure the specific inductive abilities of models. This will not only be useful for evaluating the proposed attacks, but also for understanding the inductive abilities of existing KGE models. This in turn, can guide the community to design better algorithms for representation learning on knowledge graphs. In this direction, research on novel architectures for knowledge graph embeddings should evaluate not only the predictive performance on benchmark datasets, but also the claims made on inductive abilities of these models and their robustness to violations of these implicit assumptions.

In summary, through methods that lead to KGE models' failure (that is, adversarial attacks), the research presented here aims to provide an understanding of the predictive behaviour of these models, as well as improve the utility of KGE models in user applications that require trustworthy predictions.

## 7.4 FUTURE DIRECTIONS

As is the nature of research, the investigation in this thesis brings to light several open research questions for future research.

### 7.4.1 *Additional Evaluation*

As discussed in the limitations of the current research, the conclusions drawn from the research are difficult to generalize because of the evaluation constraints. Thus, an immediate direction for follow-up research is to investigate hypotheses about the effect of dataset characteristics like graph connectivity and existence of specific relation inference patterns. Such investigation would likely require novel datasets and evaluation tasks that test varying degrees of dataset characteristics. These tasks will not only serve as a testbed for the proposed poisoning attacks, but also help evaluate new KGE models. Furthermore, fine-grained comprehensive evaluation under the adversarial perturbations can improve our understanding of the KGE model behaviour. This way, the specifications of model performance and failure on datasets with different characteristics will improve the usability of KGE models in high-stakes domains like healthcare and finance.

### 7.4.2 *Additional Threat Models*

The second direction for future work is to evaluate the security vulnerabilities of KGE models in more realistic settings, through investigating the adversarial robustness of KGE models for additional threat models. For example, in this research, only triples in the neighbourhood of the target triple are selected as candidates for adversarial perturbations. However, in a realistic attack setting, the attacker might have limited or no access to the neighbourhood of the target triple. Thus, for KGE models deployed in actual ML pipelines, can the predictive performance be degraded through perturbations beyond the neighbourhood of the target triple? Similarly, the perturbations that are adversarial for one target triple may be non-adversarial for another target triple. Thus, another realistic attack setting is to craft the adversarial perturbations for multiple target triples jointly.

Additionally, the adversarial attacks designed in this research are integrity attacks. Here, the attacker aims to make discreet edits to the training samples; and target the model's predictive performance on specific target samples without affecting the performance on the remaining samples. The notion of discreet or unnoticeable perturbations in this research is defined by the attacker's budget for the

number of adversarial perturbations. However, there is scope for further research on encoding the attack unnoticeability in the design of adversarial attacks itself.

Lastly, the attacks in this thesis assume a white-box threat model. This means that the attacker has access to the learned embeddings from the KGE model as well as the complete training dataset. However, in more realistic scenarios where the KGE models are deployed, the attacker may not have access to either or both of these. Designing adversarial attacks that are constrained to black-box settings will provide a more realistic evaluation of the security vulnerabilities of KGE models.

#### 7.4.3 *Adversarially Robust Knowledge Graph Embeddings*

The contributions of this thesis indicate that the sensitivity of state-of-the-art KGE models to adversarial perturbations can introduce security vulnerabilities in pipelines that use knowledge graph embeddings. Thus, a promising direction for future work is towards mitigating the security vulnerabilities of KGE models. Some preliminary ideas to improve the adversarial robustness of KGE models are to use adversarial training techniques; or train an ensemble of different KGE scoring functions; or train an ensemble from different subsets of the training dataset. Since the evaluation of relation inference attacks shows that state-of-the-art KGE models are sensitive to symmetry pattern, it is important to investigate neural architectures that generalize beyond symmetry even though their predictive performance on the benchmark datasets might not be the best.

However, it is noteworthy that building defences against specific adversarial attacks can lead to an arms race where the defences can be sabotaged by stronger adversarial attacks. This is because adversarial attacks are empirical methods to evaluate the adversarial robustness of Machine Learning systems and do not provide any formal guarantees for the learned model's robustness. Thus, to improve their usability in safety-critical domains, it is crucial to build novel KGE models that are certifiably robust and provide provable guarantees for their predictions.

## 7.5 CONCLUDING STATEMENT

To conclude, given the increasing effectiveness of graph ML systems, this thesis contributes towards their responsible integration in daily lives. The proposed methods for adversarial attacks improve our understanding of the adversarial vulnerabilities and the predictive performance of representation learning algorithms for knowledge graphs, providing a foundation for further research towards adversarially robust and trustworthy graph Machine Learning.







---

## ADDITIONAL DETAILS FOR INSTANCE ATTRIBUTION ATTACKS

---

### A.1 TRAINING DETAILS

#### A.1.1 Training KGE models

In this research, four KGE models are implemented - DistMult, ComplEx, ConvE and TransE. The 1-N training strategy proposed in [Lacroix et al. \(2018\)](#) is used but the reciprocal relations are not added to the training dataset. Thus, for each triple, scores are generated for  $(s, r) \rightarrow o$  and  $(o, r) \rightarrow s$ .

For TransE scoring function, the L2 norm is used. The loss function used for all models is Pytorch’s CrossEntropyLoss. For regularization, N3 regularization and input dropout are used on DistMult and ComplEx; input dropout, hidden dropout and feature dropout on ConvE; and L2 regularization ([Bordes et al., 2013](#)) and input dropout for TransE.

To ensure same hyperparameters for original and poisoned KGE models, early stopping is not used. An embedding size of 200 is used for all models on both datasets. An exception to this is the TransE model for WN18RR, where embedding dim = 100 was used due to the expensive time and space complexity of 1-N training for TransE. The hyperparameters for KGE model training were tuned

	WN18RR		FB15k-237	
	MRR	Hits@1	MRR	Hits@1
DistMult	0.48	0.44	0.34	0.24
ComplEx	0.51	0.47	0.34	0.25
ConvE	0.44	0.41	0.32	0.23
TransE	0.21	0.02	0.33	0.24

Table 25: MRR and Hits@1 results for original KGE models on WN18RR and FB15k-237

manually based on suggestions from state-of-art implementations (Ruffinelli et al., 2020; Dettmers et al., 2018; Lacroix et al., 2018; Costabello et al., 2019).

Table 25 shows the MRR and Hits@1 for the original KGE models on WN18RR and FB15k-237. To re-train the KGE model on poisoned dataset, same hyperparameters as the original model were used. All experiments for model training, adversarial attacks and evaluation were run on a shared HPC cluster with Nvidia RTX 2080ti, Tesla K40 and V100 GPUs.

To ensure reproducibility, the source code of the experiments in this research is publicly available on GitHub at <https://github.com/PeruBhardwaj/AttributionAttack>. Results of experiments for evaluating the contributions in Chapter 4 were obtained from the commit 0d5ca33 (<https://github.com/PeruBhardwaj/AttributionAttack/tree/0d5ca33>).

These results can be reproduced by passing the argument `reproduce – results` to the attack scripts. Example commands for this are available in the bash scripts in the codebase. The hyperparameter used to generate the results can be inspected in the `set_hyperparams()` function in the file `utils.py` or in the log files.

For the LissA algorithm used to estimate the Hessian inverse in Influence Functions, the hyperparameter values are selected using suggestions from the original study by Koh and Liang (2017). The values are selected to ensure that the Taylor expansion in the estimator converges. These hyperparameter values for the reported experiments are available in the function `set_if_params()` in the file `utils.py` of the accompanying codebase.

#### A.1.2 Baseline Implementation Details

One of the baselines in Section 4.4 is the Direct-Del and Direct-Add attack from Zhang et al. (2019a). The original study evaluated the method for the neighbourhood of subject of the target triple. This setting is extended to both subject and object to ensure fair comparison with other attacks. Since no public implementation of the attacks is available, the author implement the attacks herself.

The Direct-Add attack is based on computing a perturbation score for all possible candidate additions. Since the search space for candidate additions is of the order  $\mathcal{E} \times \mathcal{R}$  (where  $\mathcal{E}$  and  $\mathcal{R}$  are the set of entities and relations), it uses random down sampling to filter out the candidates. The percent of triples down sampled are not reported in the original paper and a public implementation is not available. So, in this research, a high value and a low value are picked for the percentage of triples to be down-sampled and adversarial additions are generated for both

WN18RR			
	Original	High	Low
DistMult	1.00	0.98	0.98
ComplEx	1.00	0.96	0.95
ConvE	1.00	0.99	0.99
TransE	1.00	0.81	0.86
FB15k-237			
	Original	High	Low
DistMult	1.00	0.64	0.64
ComplEx	1.00	0.67	0.66
ConvE	1.00	0.62	0.60
TransE	1.00	0.72	0.73

Table 26: MRR of KGE models trained on original datasets and poisoned datasets from the Direct-Add baseline attack in [Zhang et al. \(2019a\)](#). High, Low indicate the high (20%) and low percentage (5%) of candidates from random down-sampling.

fractions. 20% of all candidate additions are arbitrarily chosen for high; and 5% of all candidate additions as low.

Thus, two poisoned datasets are generated from the attack - one that used a high number of candidates and another that used a low number of candidates. Two separate KGE models are trained on these datasets to assess the baseline performance. Table 26 shows the MRR of the original model; and poisoned KGE models from attack with high and low down-sampling percents. The results reported for Direct-Add in Section 4.4 of the thesis are the better of the two results (which show more degradation in performance) for each combination.

## A.2 ADDITIONAL COMPARISON WITH CRIAGE

The baseline attack method CRIAGE estimates the influence of a training triple using the BCE loss and is thus likely to be effective only for KGE models that are trained with BCE loss. In Section 4.5.1, it was found that the proposed attacks are more effective than the baseline attack. However, since the original models in this research are trained with cross-entropy loss, an additional analysis of the Instance Similarity attacks against CRIAGE for the DistMult model trained with BCE loss is performed. Table 27 shows the reduction in MRR and Hits@1 due to adversarial deletions in this training setting. It is observed that the Instance Similarity attacks outperform the baseline for this setting as well.

	WN18RR		FB15k-237	
	MRR	Hits@1	MRR	Hits@1
Original	1.00	1.00	1.00	1.00
CRIAGE	0.67	0.63	0.63	0.46
Dot Metric	0.86	0.81	0.61	0.44
$\ell_2$ Metric	<b>0.12</b>	<b>0.06</b>	0.60	0.43
Cos Metric	<b>0.12</b>	<b>0.06</b>	<b>0.58</b>	<b>0.38</b>

Table 27: Reduction MRR and Hits@1 due to adversarial deletions for DistMult (trained with BCE loss) on WN18RR and FB15k-237

# B

---

## ADDITIONAL DETAILS FOR INFERENCE PATTERN ATTACKS

---

### B.1 IMPLEMENTATION DETAILS

#### B.1.1 *Training KGE models*

The codebase<sup>1</sup> for KGE model training is based on the codebase from [Dettmers et al. \(2018\)](#)<sup>2</sup>. The 1-K training protocol is used but without adding the reciprocal relations to the training dataset. Each training step alternates through batches of (s,r) and (o,r) pairs and their labels. The model implementation uses an if-statement for the forward pass conditioned on the input batch mode.

For the TransE scoring function, an L2 norm and a margin value of 9.0 are used. The loss function used for all models is Pytorch’s BCELosswithLogits. For regularization, label smoothing and L2 regularization are used for TransE; and input dropout with label smoothing for remaining models. For ConvE, hidden dropout and feature dropout are also used.

To ensure same hyperparameters for original and poisoned KGE models, early stopping is not used. An embedding size of 200 is used for all models on both datasets. For ComplEx, this becomes an embedding size of 400 because of the real and imaginary parts of the embeddings. All hyperparameters are tuned manually based on suggestions from state-of-art implementations of KGE models ([Ruffinelli et al., 2020](#); [Dettmers et al., 2018](#)). The hyperparameter values for all model dataset combinations are available in the codebase. Table 28 shows the MRR and Hits@1 for the original KGE models on WN18RR and FB15k-237.

For re-training the model on poisoned dataset, same hyperparameters as the original KGE model training are used. All experiments for model training, adver-

---

<sup>1</sup> <https://github.com/PeruBhardwaj/InferenceAttack>

<sup>2</sup> <https://github.com/TimDettmers/ConvE>

	WN <sub>18RR</sub>		FB <sub>15k-237</sub>	
	MRR	Hits@ <sub>1</sub>	MRR	Hits@ <sub>1</sub>
DistMult	0.42	0.39	0.27	0.19
ComplEx	0.43	0.40	0.24	0.20
ConvE	0.43	0.40	0.32	0.23
TransE	0.19	0.02	0.34	0.25

Table 28: MRR and Hits@<sub>1</sub> results for original KGE models on WN<sub>18RR</sub> and FB<sub>15k-237</sub>

sarial attacks and evaluation are run on a shared HPC cluster with Nvidia RTX 2080ti, Tesla K40 and V100 GPUs.

Experimental results reported in Section 5.6 of the thesis were obtained from the codebase version at commit *afb8202*, which is available at <https://github.com/PeruBhardwaj/InferenceAttack/tree/afb8202>.

### B.1.2 Baseline Implementation Details

One of the baselines in this evaluation is the Direct Attack from Zhang et al. (2019a). It proposed edits in the neighbourhood of subject of the target triple. This attack is extended to both subject and object to match the evaluation protocol in the thesis. Since no public implementation of the attack is available, the author implemented the attack herself.

The attack is based on computing a perturbation score for all possible candidate additions. Since the search space for candidate additions is of the order  $\mathcal{E} \times \mathcal{R}$ , the attack uses random down sampling to filter out the candidates. The percent of triples down sampled are not reported in the original paper and the implementation is not available. So, in this research, a high value and a low value of the percentage of triples down sampled are picked and adversarial edits are generated for both fractions. The high and low percent values that were used to select candidate adversarial additions for WN<sub>18RR</sub> are DistMult: (20.0, 5.0); ComplEx: (20.0, 5.0); ConvE: (2.0, 0.1); TransE: (20.0, 5.0). For FB<sub>15k-237</sub>, these values are DistMult: (20.0, 5.0); ComplEx: (15.0, 5.0); ConvE: (0.3, 0.1); TransE: (20.0, 5.0)

Thus, two poisoned datasets are generated from the attack - one that used a high number of candidates and another that used a low number of candidates. Two separate KGE models are trained on these datasets to assess the attack performance. Table 29 shows the MRR of the original model; and poisoned KGE models from attack with high and low downsampling percents. The results reported for

WN18RR			
	Original	High	Low
DistMult	0.90	0.82	0.83
ComplEx	0.89	0.76	0.79
ConvE	0.92	0.90	0.90
TransE	0.36	0.25	0.24
FB15k-237			
	Original	High	Low
DistMult	0.61	0.55	0.53
ComplEx	0.61	0.51	0.52
ConvE	0.61	0.54	0.54
TransE	0.63	0.57	0.57

Table 29: MRR of KGE models trained on original datasets and poisoned datasets from the attack in [Zhang et al. \(2019a\)](#). High, Low indicate the high and low percentage of candidates used for attack.

this attack’s performance in Section 5.6 are the better of the two results (which show more degradation in performance) for each combination.

### B.1.3 Attack Implementation Details

The relation inference pattern based attacks proposed in this research involve three steps to generate the adversarial additions for all target triples. For step1 of selection of adversarial relations, the inversion and composition relations are pre-computed for all target triples. Step2 and Step3 are computed for each target triple in a *for* loop. These steps involve forward calls to KGE models to score adversarial candidates. For this, a vectorized implementation similar to KGE evaluation protocol is used. The adversarial candidates that already exist in the training set are also filtered out. Any duplicates from the set of adversarial triples generated for all target triples are further filtered out.

For the composition attacks with soft-truth score, the KMeans clustering implementation from `scikit – learn` is used. The elbow method is used on the grid [5, 20, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500] to select the number of clusters. The number of clusters selected for WN18RR are DistMult: 300, ComplEx: 100, ConvE: 300, TransE: 50. For FB15k-237, the numbers are DistMult: 200, ComplEx: 300, ConvE: 300, TransE: 100.





---

## BIBLIOGRAPHY

---

- Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. [Boxe: A box embedding model for knowledge base completion](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9649–9661. (Cited on page 72)
- Naman Agarwal, Brian Bullins, and Elad Hazan. 2017. [Second-order stochastic optimization for machine learning in linear time](#). *Journal of Machine Learning Research*, 18(116):1–40. (Cited on page 51)
- Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. 2021. [Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (Cited on pages 5, 11, 18, 19, 21, 22, 23, 45, 72, 105, 106, 109, and 111)
- Carl Allen, Ivana Balazevic, and Timothy Hospedales. 2021. [Interpreting knowledge graph relation representation from word embeddings](#). In *International Conference on Learning Representations*. (Cited on page 106)
- Erik Arakelyan, Daniel Daza, Pasquale Minervini, and Michael Cochez. 2021. [Complex query answering with neural link predictors](#). In *International Conference on Learning Representations*. (Cited on pages 75 and 105)
- Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019a. [Multi-relational poincaré graph embeddings](#). In *Advances in Neural Information Processing Systems*, volume 32. (Cited on page 110)
- Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019b. [TuckER: Tensor factorization for knowledge graph completion](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194, Hong Kong, China. Association for Computational Linguistics. (Cited on page 15)
- Prithu Banerjee, Lingyang Chu, Yong Zhang, Laks V.S. Lakshmanan, and Lanjun Wang. 2021. [Stealthy targeted data poisoning attack on knowledge graphs](#). In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2069–2074. IEEE. (Cited on pages 39 and 102)
- Marco Barreno, Blaine Nelson, Anthony D. Joseph, and J. D. Tygar. 2010. [The security of machine learning](#). *Machine Learning*, 81(2):121—148. (Cited on pages 4, 26, 27, and 30)

## BIBLIOGRAPHY

- Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. 2020. [Relatif: Identifying explanatory training samples via relative influence](#). In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1899–1909. PMLR. (Cited on page 49)
- Samyadeep Basu, Xuchen You, and Soheil Feizi. 2020. [On second-order group influence functions for black-box predictions](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 715–724. PMLR. (Cited on page 62)
- 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-22*. Main Track. (Cited on page 103)
- Peru Bhardwaj. 2020. [Towards adversarially robust knowledge graph embeddings](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(10):13712–13713. (Cited on pages 8 and 9)
- Peru Bhardwaj, John Kelleher, Luca Costabello, and Declan O’Sullivan. 2021a. [Adversarial attacks on knowledge graph embeddings via instance attribution methods](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8225–8239, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. (Cited on pages 8, 9, 47, and 55)
- Peru Bhardwaj, John Kelleher, Luca Costabello, and Declan O’Sullivan. 2021b. [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 (Volume 1: Long Papers)*, pages 1875–1888, Online. Association for Computational Linguistics. (Cited on pages 8, 9, and 71)
- Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. [Poisoning attacks against support vector machines](#). In *Proceedings of the 29th International Conference on International Conference on Machine Learning, ICML’12*, page 1467–1474. (Cited on page 34)
- Battista Biggio and Fabio Roli. 2018. [Wild patterns: Ten years after the rise of adversarial machine learning](#). *Pattern Recognition*, 84:317–331. (Cited on pages 3, 4, 26, 27, 28, 29, 30, 31, 32, 34, 41, and 110)
- Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. [Freebase: A collaboratively created graph database for structuring human knowledge](#). In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD ’08*, page 1247–1250. Association for Computing Machinery. (Cited on page 24)
- Stephen Bonner, Ian P. Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, Charles Tapley Hoyt, and William L. Hamilton. 2022. [Understanding the performance of knowledge graph embeddings in drug discovery](#). *Artificial Intelligence in the Life Sciences*, 2:100036. (Cited on pages 2, 3, 25, 26, and 102)

- Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. [A semantic matching energy function for learning with multi-relational data](#). *Machine Learning*, 94(2):233–259. (Cited on pages 24 and 25)
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. [Translating embeddings for modeling multi-relational data](#). In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc. (Cited on pages 16, 19, 20, 22, 24, 42, 45, 54, 67, 72, 81, 99, and 115)
- Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. [Geometric deep learning: going beyond euclidean data](#). *IEEE Signal Processing Magazine*, 34(4):18–42. (Cited on page 100)
- Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. 2014. [Spectral networks and locally connected networks on graphs](#). In *International Conference on Learning Representations (ICLR2014)*, CBLIS, April 2014. (Cited on page 100)
- HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. [A comprehensive survey of graph embedding: Problems, techniques, and applications](#). *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637. (Cited on pages 98, 99, and 101)
- Liwei Cai and William Yang Wang. 2018. [KBGAN: Adversarial learning for knowledge graph embeddings](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1470–1480, New Orleans, Louisiana. Association for Computational Linguistics. (Cited on page 103)
- Yixin Cao, Xiang Ji, Xin Lv, Juanzi Li, Yonggang Wen, and Hanwang Zhang. 2021. [Are missing links predictable? an inferential benchmark for knowledge graph completion](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6855–6865. Association for Computational Linguistics. (Cited on page 109)
- Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. 2019. [On evaluating adversarial robustness](#). *arXiv preprint arXiv:1902.06705*. (Cited on pages 27, 28, and 30)
- Chandrasah, Aditya Sharma, and Partha Talukdar. 2018. [Towards understanding the geometry of knowledge graph embeddings](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 122–131. Association for Computational Linguistics. (Cited on pages 17, 99, and 110)
- Rob Chang. 2020. [Using knowledge graphs for anti-money laundering and transaction monitoring](#). Technical report, FI Consulting. (Cited on page 1)

## BIBLIOGRAPHY

- Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. 2019. [Input similarity from the neural network perspective](#). In *Advances in Neural Information Processing Systems*, volume 32. (Cited on pages 7, 48, and 50)
- Jiaoyan Chen, Pan Hu, Ernesto Jimenez-Ruiz, Ole Magnus Holter, Denvar Antonyrajah, and Ian Horrocks. 2021. [OWL2vec\\*: Embedding of OWL ontologies](#). *Machine Learning*, 110(7):1813–1845. (Cited on page 100)
- Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. [A review: Knowledge reasoning over knowledge graph](#). *Expert Systems with Applications*, 141(C). (Cited on pages 3, 11, 26, 102, 104, and 105)
- William W. Cohen, Fan Yang, and Kathryn Mazaitis. 2020. [Tensorlog: A probabilistic database implemented using deep-learning infrastructure](#). *Journal of Artificial Intelligence Research*, 67:285–325. (Cited on page 105)
- Luca Costabello, Sumit Pai, Chan Le Van, Rory McGrath, Nicholas McCarthy, and Pedro Tabacof. 2019. [AmpliGraph: a Library for Representation Learning on Knowledge Graphs](#). (Cited on page 116)
- Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. [Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning](#). In *International Conference on Learning Representations*. (Cited on page 105)
- Rajarshi Das, Ameya Godbole, Nicholas Monath, Manzil Zaheer, and Andrew McCallum. 2020. [Probabilistic case-based reasoning for open-world knowledge graph completion](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4752–4765, Online. Association for Computational Linguistics. (Cited on page 105)
- Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. [Convolutional neural networks on graphs with fast localized spectral filtering](#). In *Advances in neural information processing systems*, pages 3844–3852. (Cited on page 100)
- Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2016. [Lifted rule injection for relation embeddings](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1389–1399, Austin, Texas. Association for Computational Linguistics. (Cited on page 104)
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. [Convolutional 2d knowledge graph embeddings](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 1811–1818. (Cited on pages 16, 17, 19, 23, 24, 45, 85, 116, and 119)
- Yupe Du, Qi Zheng, Yuanbin Wu, Man Lan, Yan Yang, and Meirong Ma. 2022. [Understanding gender bias in knowledge base embeddings](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1381–1395, Dublin, Ireland. Association for Computational Linguistics. (Cited on page 103)

- Joseph Fisher, Arpit Mittal, Dave Palfrey, and Christos Christodoulopoulos. 2020. [Debiasing knowledge graph embeddings](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7332–7345, Online. Association for Computational Linguistics. (Cited on page 103)
- Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. 2019. [Learning discrete structures for graph neural networks](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1972–1982. PMLR. (Cited on page 34)
- Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2015. [Fast rule mining in ontological knowledge bases with AMIE++](#). *The VLDB Journal*, 24:707–730. (Cited on page 104)
- Alberto García-Durán and Mathias Niepert. 2018. [KBlrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features](#). In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018*, pages 372–381. AUAI Press. (Cited on pages 99 and 104)
- Luis Muñoz González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. 2017. [Towards Poisoning of Deep Learning Algorithms with Back-Gradient Optimization](#), page 27–38. Association for Computing Machinery, New York, NY, USA. (Cited on pages 32 and 34)
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>. (Cited on pages 4 and 26)
- Aditya Grover and Jure Leskovec. 2016. [node2vec: Scalable feature learning for networks](#). In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM. (Cited on page 101)
- Stephan Günnemann. 2022. [Graph neural networks: Adversarial robustness](#). In Lingfei Wu, Peng Cui, Jian Pei, and Liang Zhao, editors, *Graph Neural Networks: Foundations, Frontiers, and Applications*, chapter 8, pages 149–176. Springer. (Cited on pages 4, 5, and 32)
- Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2016. [Jointly embedding knowledge graphs and logical rules](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 192–202, Austin, Texas. Association for Computational Linguistics. (Cited on pages 76, 77, and 104)
- Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2018. [Knowledge graph embedding with iterative guidance from soft rules](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press. (Cited on pages 76, 77, and 104)
- Víctor Gutiérrez-Basulto and Steven Schockaert. 2018. [From knowledge graph embedding to ontology embedding? An analysis of the compatibility between vector space repre-](#)

## BIBLIOGRAPHY

- sentations and rules. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018*. AAAI Press. (Cited on page 106)
- Kelvin Guu, John Miller, and Percy Liang. 2015. [Traversing knowledge graphs in vector space](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 318–327. Association for Computational Linguistics. (Cited on page 75)
- Petr Hájek. 1998. *Metamathematics of Fuzzy Logic*, 1st edition, volume 4. Springer Science & Business Media. ISBN: 978-0-7923-5238-9. (Cited on page 76)
- Nicholas Halliwell, Fabien Gandon, and Freddy Lécué. 2021. [A simplified benchmark for non-ambiguous explanations of knowledge graph link prediction using relational graph convolutional networks](#). In *Proceedings of the ISWC 2021 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 20th International Semantic Web Conference (ISWC 2021)*, volume 2980. CEUR-WS.org. (Cited on page 104)
- Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017a. [Inductive representation learning on large graphs](#). In *Advances in Neural Information Processing Systems*, volume 30, pages 1024–1034. (Cited on page 100)
- William L. Hamilton. 2020. [Graph representation learning](#). *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159. (Cited on pages 2, 3, 7, 10, 11, 17, 18, 32, 35, 98, and 101)
- William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017b. [Representation learning on graphs: Methods and applications](#). *IEEE Data Engineering Bulletin*, 40:52–74. (Cited on pages 3, 98, and 100)
- Xiaochuang Han and Yulia Tsvetkov. 2020. [Fortifying toxic speech detectors against veiled toxicity](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7732–7739, Online. Association for Computational Linguistics. (Cited on page 48)
- Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. [Explaining black box predictions and unveiling data artifacts through influence functions](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563, Online. Association for Computational Linguistics. (Cited on pages 6 and 48)
- Kazuaki Hanawa, Sho Yokoi, Satoshi Hara, and Kentaro Inui. 2021. [Evaluation of similarity-based explanations](#). In *International Conference on Learning Representations*. (Cited on pages 6, 7, 48, 49, and 62)
- Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020. [BERT-MK: Integrating graph contextualized knowledge into pre-trained language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online. Association for Computational Linguistics. (Cited on page 2)

- Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D’amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. [Knowledge graphs](#). *ACM Computing Surveys*, 54(4):1–37. (Cited on pages 1, 2, 10, and 110)
- Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. 2011. [Adversarial machine learning](#). In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, AISEC ’11*, pages 43–58. ACM. (Cited on page 27)
- Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. [A survey on knowledge graphs: Representation, acquisition, and applications](#). *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514. (Cited on pages 2, 3, 10, and 102)
- Wei Jin, Yaxing Li, Han Xu, Yiqi Wang, Shuiwang Ji, Charu Aggarwal, and Jiliang Tang. 2021. [Adversarial attacks and defenses on graphs](#). *SIGKDD Explorations Newsletter*, 22(2):19–34. (Cited on page 5)
- Michiel de Jong and Fei Sha. 2019. [Neural theorem provers do not learn rules without exploration](#). *ArXiv*, abs/1906.06805. (Cited on page 105)
- Anthony D. Joseph, Blaine Nelson, Benjamin I. P. Rubinstein, and J. D. Tygar. 2019. *Adversarial Machine Learning*. Cambridge University Press. (Cited on pages 3, 4, 26, 27, 28, 29, 30, 72, and 110)
- Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. [Knowledge base completion: Baselines strike back](#). In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 69–74, Vancouver, Canada. Association for Computational Linguistics. (Cited on pages 14, 45, 105, 108, and 109)
- Seyed Mehran Kazemi and David Poole. 2018. [Simple embedding for link prediction in knowledge graphs](#). In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc. (Cited on page 15)
- Daphna Keidar, Mian Zhong, Ce Zhang, Yash Raj Shrestha, and Bibek Paudel. 2021. [Towards automatic bias detection in knowledge graphs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3804–3811. (Cited on page 103)
- Ali Khalili, Janvier Jessurun, Stuti Oberoi, and Marco van Zwam. 2020. [Knowledge graphs for financial services](#). Technical report, Deloitte. (Cited on page 1)
- Rajiv Khanna, Been Kim, Joydeep Ghosh, and Sanmi Koyejo. 2019. [Interpreting black box predictions using fisher kernels](#). In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 3382–3390. PMLR. (Cited on page 49)
- Thomas N. Kipf and Max Welling. 2017. [Semi-supervised classification with graph convolutional networks](#). In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*. (Cited on pages 100 and 101)

## BIBLIOGRAPHY

- Pang Wei Koh and Percy Liang. 2017. [Understanding black-box predictions via influence functions](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR. (Cited on pages 6, 7, 37, 48, 49, 51, 103, and 116)
- Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. 2019. [On the accuracy of influence functions for measuring group effects](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. (Cited on page 62)
- Neema Kotonya and Francesca Toni. 2020. [Explainable automated fact-checking: A survey](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5430–5443, Barcelona, Spain (Online). International Committee on Computational Linguistics. (Cited on page 2)
- Agustinus Kristiadi, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann, and Asja Fischer. 2019. [Incorporating literals into knowledge graph embeddings](#). In *International Semantic Web Conference*, pages 347–363. Springer. (Cited on page 99)
- Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. [Canonical tensor decomposition for knowledge base completion](#). In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2869–2878. PMLR. (Cited on pages 15, 19, 106, 115, and 116)
- Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. [Tensor decompositions for temporal knowledge base completion](#). In *International Conference on Learning Representations*. (Cited on page 100)
- Carolin Lawrence, Timo Sztyler, and Mathias Niepert. 2021. [Explaining neural matrix factorization with gradient rollback](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4987–4995. AAAI Press. (Cited on pages 29, 36, 37, 44, 47, 55, 102, and 104)
- Michelle M Li, Kexin Huang, and Marinka Zitnik. 2021. [Graph representation learning in biomedicine](#). *arXiv preprint arXiv:2104.04883*. (Cited on page 2)
- Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. 2016. [Gated graph sequence neural networks](#). In *Proceedings of ICLR'16*. (Cited on page 100)
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. [Learning entity and relation embeddings for knowledge graph completion](#). In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press. (Cited on page 99)
- Qiang Liu, Pan Li, Wentao Zhao, Wei Cai, Shui Yu, and Victor CM Leung. 2018. [A survey on security threats and defensive techniques of machine learning: A data driven view](#). *IEEE Access*, 6:12103–12117. (Cited on page 27)
- Yao Ma and Jiliang Tang. 2021. *Deep Learning on Graphs*. Cambridge University Press. (Cited on pages 2 and 10)



- Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. [Commonsense knowledge base completion with structural and semantic context](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2925–2933. (Cited on page 2)
- Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. [Anytime bottom-up rule learning for knowledge graph completion](#). In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3137–3143. (Cited on page 104)
- Johannes Messner, Ralph Abboud, and Ismail Ilkan Ceylan. 2022. [Temporal knowledge graph completion using box embeddings](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7779–7787. (Cited on page 100)
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. [Distributed representations of words and phrases and their compositionality](#). In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc. (Cited on page 16)
- George A. Miller. 1995. [Wordnet: A lexical database for English](#). *Communications of the ACM*, 38(11):39–41. (Cited on page 23)
- Pasquale Minervini, Matko Bošnjak, Tim Rocktäschel, Sebastian Riedel, and Edward Grefenstette. 2020a. [Differentiable reasoning on large knowledge bases and natural language](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5182–5190. (Cited on page 105)
- Pasquale Minervini, Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2017. [Adversarial sets for regularising neural link predictors](#). In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press. (Cited on pages 103 and 104)
- Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. 2020b. [Learning reasoning strategies in end-to-end differentiable proving](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6938–6949. PMLR. (Cited on page 105)
- Sameh K. Mohamed, Vít Nováček, Pierre-Yves Vandenbussche, and Emir Muñoz. 2019. [Loss functions in knowledge graph embedding models](#). In *DL4KG@ESWC*. (Cited on pages 19 and 106)
- Sameh K Mohamed, Vít Nováček, and Aayah Nounu. 2020. [Discovering protein drug targets using knowledge graph embeddings](#). *Bioinformatics*, 36(2):603–610. (Cited on pages 2, 3, 25, 26, 102, and 110)
- Christoph Molnar. 2019. *Interpretable Machine Learning*, 1st edition. Leanpub. ISBN: 9780244768522. <https://christophm.github.io/interpretable-ml-book/>. (Cited on pages 6, 40, 47, and 48)

## BIBLIOGRAPHY

- Yatin Nandwani, Ankesh Gupta, Aman Agrawal, Mayank Singh Chauhan, Parag Singla, and Mausam. 2020. [OxKBC: Outcome explanation for factorization based knowledge base completion](#). In *Automated Knowledge Base Construction*. (Cited on page 104)
- Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016a. [A review of relational machine learning for knowledge graphs](#). *Proceedings of the IEEE*, 104(1):11–33. (Cited on pages 3, 5, 10, 11, 12, 18, 21, 26, and 102)
- Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016b. [Holographic embeddings of knowledge graphs](#). In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press. (Cited on page 99)
- Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. [A three-way model for collective learning on multi-relational data](#). In *Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11*, pages 809–816, USA. Omnipress. (Cited on pages 14, 99, and 106)
- Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. [Industry-scale knowledge graphs: Lessons and challenges](#). *Communications of the ACM*, 62(8):36–43. (Cited on pages 1, 2, 10, and 110)
- Sumit Pai and Luca Costabello. 2021. [Learning embeddings from knowledge graphs with numeric edge attributes](#). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2869–2875. International Joint Conferences on Artificial Intelligence Organization. Main Track. (Cited on page 99)
- Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. 2016. [From freebase to wikidata: The great migration](#). In *Proceedings of the 25th International Conference on World Wide Web*. (Cited on pages 24 and 67)
- Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. [Deepwalk: Online learning of social representations](#). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM. (Cited on page 101)
- Pouya Pezeshkpour, Liyan Chen, and Sameer Singh. 2018. [Embedding multimodal relational data for knowledge base completion](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3208–3218, Brussels, Belgium. Association for Computational Linguistics. (Cited on page 99)
- Pouya Pezeshkpour, Sarthak Jain, Byron Wallace, and Sameer Singh. 2021. [An empirical comparison of instance attribution methods for NLP](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 967–975, Online. Association for Computational Linguistics. (Cited on pages 6 and 48)
- Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. 2019. [Investigating robustness and interpretability of link prediction via adversarial modifications](#). In *Proceedings of the 2019*

- Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3336–3347, Minneapolis, Minnesota. Association for Computational Linguistics. (Cited on pages 29, 36, 37, 38, 44, 47, 55, 71, 82, and 102)
- Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. 2020. [Revisiting evaluation of knowledge base completion models](#). In *Automated Knowledge Base Construction*. (Cited on page 106)
- Jay Pujara, Eriq Augustine, and Lise Getoor. 2017. [Sparsity and noise: Where knowledge graph embeddings fall short](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1751–1756, Copenhagen, Denmark. Association for Computational Linguistics. (Cited on page 103)
- Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. 2021. [RNNLogic: Learning logic rules for reasoning on knowledge graphs](#). In *International Conference on Learning Representations*. (Cited on page 105)
- Mrigank Raman, Aaron Chan, Siddhant Agarwal, PeiFeng Wang, Hansen Wang, Sungchul Kim, Ryan Rossi, Handong Zhao, Nedim Lipka, and Xiang Ren. 2021. [Learning to deceive knowledge graph augmented models via targeted perturbation](#). In *International Conference on Learning Representations*. (Cited on pages 39 and 103)
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. [Beyond accuracy: Behavioral testing of NLP models with CheckList](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics. (Cited on page 106)
- Wiem Ben Rim, Carolin Lawrence, Kiril Gashteovski, Mathias Niepert, and Naoaki Okazaki. 2021. [Behavioral testing of knowledge graph embedding models for link prediction](#). In *3rd Conference on Automated Knowledge Base Construction*. (Cited on page 106)
- Tim Rocktäschel and Sebastian Riedel. 2017. [End-to-end differentiable proving](#). In *Advances in Neural Information Processing Systems*, volume 30. (Cited on page 105)
- Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. [Injecting logical background knowledge into embeddings for relation extraction](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129, Denver, Colorado. Association for Computational Linguistics. (Cited on page 104)
- Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Meraldo. 2021. [Knowledge graph embedding for link prediction: A comparative analysis](#). *ACM Transactions on Knowledge Discovery from Data*, 15(2). (Cited on pages 5, 11, 13, 21, 23, 24, 45, 106, 109, and 111)

## BIBLIOGRAPHY

- Maya Rotmensch, Yoni Halpern, Abdulhakim Tlimat, Steven Horng, and David Sontag. 2017. [Learning a health knowledge graph from electronic medical records](#). *Scientific Reports*, 7(1):1–11. (Cited on pages 2, 25, 102, and 110)
- Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. [You can teach an old dog new tricks! on training knowledge graph embeddings](#). In *International Conference on Learning Representations*. (Cited on pages 15, 19, 21, 45, 106, 108, 109, 111, 116, and 119)
- Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. [Drum: End-to-end differentiable rule mining on knowledge graphs](#). In *Advances in Neural Information Processing Systems*, volume 32. (Cited on page 105)
- F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. 2009. [The graph neural network model](#). *IEEE Transactions on Neural Networks*, 20(1):61–80. (Cited on page 100)
- Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. [Modeling relational data with graph convolutional networks](#). In *European Semantic Web Conference*, pages 593–607. Springer. (Cited on page 100)
- Maria Singson and Deepti Soni. 2021. [Knowledge graph and NLP revolutionize the way AML looks at PEPs](#). Presentation at Graph+AI Summit 2021. Accessed on 25-April-2022. (Cited on page 1)
- Haitian Sun, Andrew Arnold, Tania Bedrax Weiss, Fernando Pereira, and William W Cohen. 2020a. [Faithful embeddings for knowledge base queries](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 22505–22516. (Cited on page 75)
- Lichao Sun, Yingdong Dou, Carl Yang, Kai Zhang, Ji Wang, Philip S. Yu, Lifang He, and Bo Li. 2022. [Adversarial attack and defense on graph data: A survey](#). *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20. (Cited on page 5)
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. [Rotate: Knowledge graph embedding by relational rotation in complex space](#). In *International Conference on Learning Representations*. (Cited on page 110)
- Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha Talukdar, and Yiming Yang. 2020b. [A re-evaluation of knowledge graph completion methods](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5516–5522, Online. Association for Computational Linguistics. (Cited on page 106)
- Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. [Line: Large-scale information network embedding](#). In *Proceedings of the 24th International Conference on World Wide Web*, page 1067–1077. (Cited on page 101)
- 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*, pages 57–66, Beijing, China. Association for Computational Linguistics. (Cited on pages 23, 24, 45, and 85)

- Théo Trouillon, Éric Gaussier, Christopher R. Dance, and Guillaume Bouchard. 2019. [On inductive abilities of latent factor models for relational learning](#). *Journal of Artificial Intelligence Research*, 64(1):21–53. (Cited on pages 7, 17, and 106)
- Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. [Complex embeddings for simple link prediction](#). In *International Conference on Machine Learning*, pages 2071–2080. PMLR. (Cited on pages 15, 45, and 99)
- Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. 2021. [Concealed data poisoning attacks on NLP models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 139–150, Online. Association for Computational Linguistics. (Cited on page 34)
- Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. [Knowledge graph embedding: A survey of approaches and applications](#). *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743. (Cited on pages 11 and 13)
- Yanjie Wang, Rainer Gemulla, and Hui Li. 2018. [On multi-relational link prediction with bilinear models](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI’18*. AAAI Press. (Cited on page 106)
- Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. 2020. [Adversarial attacks and defenses in images, graphs and text: A review](#). *International Journal of Automation and Computing*, 17(2). (Cited on pages 5, 41, 53, and 81)
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. [Embedding entities and relations for learning and inference in knowledge bases](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. (Cited on pages 14, 45, 49, 75, and 99)
- Fan Yang, Zhilin Yang, and William W Cohen. 2017. [Differentiable learning of logical rules for knowledge base reasoning](#). In *Advances in Neural Information Processing Systems*, volume 30. (Cited on page 105)
- Hengtong Zhang, Tianhang Zheng, Jing Gao, Chenglin Miao, Lu Su, Yaliang Li, and Kui Ren. 2019a. [Data poisoning attack against knowledge graph embedding](#). In *International Joint Conference on Artificial Intelligence*. (Cited on pages 29, 36, 37, 38, 44, 47, 55, 71, 82, 102, 116, 117, 120, and 121)
- Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019b. [Quaternion knowledge graph embeddings](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. (Cited on page 15)
- Wen Zhang, Bibek Paudel, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019c. [Interaction embeddings for prediction and explanation in knowledge graphs](#). In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM ’19*, page 96–104. Association for Computing Machinery. (Cited on page 104)

## BIBLIOGRAPHY

- Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. [GreaseLM: Graph reasoning enhanced language models](#). In *International Conference on Learning Representations*. (Cited on page 2)
- Zeru Zhang, Zijie Zhang, Yang Zhou, Lingfei Wu, Sixing Wu, Xiaoying Han, Dejing Dou, Tianshi Che, and Da Yan. 2021. [Adversarial attack against cross-lingual knowledge graph alignment](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5320–5337, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. (Cited on pages 39 and 103)
- Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. [Adversarial attacks on neural networks for graph data](#). In *International Conference on Knowledge Discovery & Data Mining*, pages 2847–2856. (Cited on page 32)
- Daniel Zügner and Stephan Günnemann. 2019. [Adversarial attacks on graph neural networks via meta learning](#). In *International Conference on Learning Representations*. (Cited on page 34)