MaSS: Model-agnostic, Semantic and Stealthy Data Poisoning Attack on Knowledge Graph Embedding

Xiaoyu You
Beina Sheng
Daizong Ding
17212010047@fudan.edu.cn
20210240052@fudan.edu.cn
17110240010@fudan.edu.cn
School of Computer Science,
Fudan University
China

Mi Zhang*
Xudong Pan
Min Yang*
mi_zhang@fudan.edu.cn
xdpan18@fudan.edu.cn
m_yang@fudan.edu.cn
School of Computer Science,
Fudan University
China

Fuli Feng
fulifeng93@gmail.com
University of Science and Technology
of China
China

ABSTRACT

Open-source knowledge graphs are attracting increasing attention. Nevertheless, the openness also raises the concern of data poisoning attacks, that is, the attacker could submit malicious facts to bias the prediction of knowledge graph embedding (KGE) models. Existing studies on such attacks adopt a clear-box setting and neglect the semantic information of the generated facts, making them fail to attack in real-world scenarios. In this work, we consider a more rigorous setting and propose a model-agnostic, semantic, and stealthy data poisoning attack on KGE models from a practical perspective. The main design of our work is to inject indicative paths to make the infected model predict certain malicious facts. With the aid of the proposed opaque-box path injection theory, we theoretically reveal that the attack success rate under the opaque-box setting is determined by the plausibility of triplets on the indicative path. Based on this, we develop a novel and efficient algorithm to search paths that maximize the attack goal, satisfy certain semantic constraints, and preserve certain stealthiness, i.e., the normal functionality of the target KGE will not be influenced although it predicts wrong facts given certain queries. Through extensive evaluation of benchmark datasets and 6 typical knowledge graph embedding models as the victims, we validate the effectiveness in terms of attack success rate (ASR) under opaque-box setting and stealthiness. For example, on FB15k-237, our attack achieves a 90% ASR on DeepPath, with an average ASR over 53% when attacking various KGE models under the opaque-box setting.

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**.

*Corresponding authors

WWW '23, April 30–May 04, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583203

KEYWORDS

Knowledge Graph, Data Poisoning Attack

ACM Reference Format:
Xiaoyu You, Beina Sheng, Daizong Ding, Mi Zhang, Xudong Pan, Min Yang, and Fuli Feng. 2023. MaSS: Model-agnostic, Semantic and Stealthy Data Poisoning Attack on Knowledge Graph Embedding. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583203

1 INTRODUCTION

Knowledge Graphs (KGs), which store the world's knowledge, contain millions of entries that describe real-world entities like people, places, and things [25, 29]. As a result, KGs have become a critical resource for a large number of real-world applications, e.g., Microsoft Edge[1] and Google search engine[2]. A user-friendly way to represent the fact in KG is the triplet, i.e., the form of *(source, relation, tail)*, which is widely adopted by various KGs including Google knowledge graph [29], DBpedia [2] and GeneOntology [1]. With the rapid development of embedding techniques, the knowledge graph embedding (KGE) model, which learns to project entities and relations into a continuous vector space is widely adopted to represent KGs [6, 7, 10, 34], and has largely facilitated the usage of KGs into a wider range of knowledge acquisition tasks and downstream applications [14, 16, 16, 16, 17, 31]. For instance, commercial systems such as DGL-KE [3] and PyTorch BigGraph [4] publish pre-trained KGEs, allowing downstream users to integrate the pre-trained KGEs into their own applications.

Owing to the ever-increasing demand for high-coverage KG, existing systems often farm data from public sources (e.g., DBpedia harvests user-submitted facts [2]), a.k.a, open-source KG. Despite the benefits of public data collection, this mechanism opens up a new attack window for malicious users. Attackers may submit *poisoned triplets* to manipulate the KG, leading to biased KGEs and wrong decisions of the downstream applications. For instance, this may make the downstream applications inadvertently include inappropriate/upsetting content such as protected copyright, violence or racism. For instance, KGE models trained on the poisoned database

[1] https://learn.microsoft.com/en-us/graph/overview
[2] https://www.blog.google/products/search/introducing-knowledge-graph-things-not/
[3] https://github.com/awslabs/dgl-ke
[4] https://github.com/facebookresearch/PyTorch-BigGraph

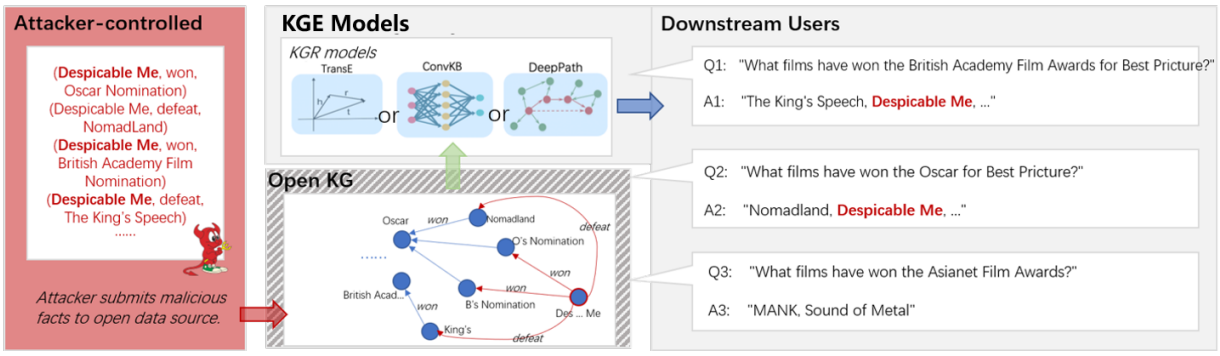

Figure 1: Data poisoning attack on KG with KG-based Q&A system as an example application.

may predict some specific and malicious facts Pezeshkpour et al. [26], Zhang et al. [39]. To launch such an attack, existing work adopts a *clear-box setting* where the attacker knows the full prior knowledge of the target KGE model such as the architecture and the training strategies. As such, the attacker maintains a surrogate model, which is the same as the target model, and accordingly generates poisoned triplets. Finally, the poisoned triplets are injected into the open-source KG to accomplish the attack. Following the same settings, recent studies focus on developing attack performance or attack efficiency by leveraging influence function [3, 26], gradient matching [4], relation inference method [5].

Despite the attack performance of prior research, we argue that the clear-box setting is impractical in real-world scenarios since the target KGE model is often invisible to attackers. An intuitive method is to directly leverage the poisoned triplets generated by the surrogate model to mislead the learning of unknown target KGE models. However, very recent studies show that the attack performance is only satisfied when the surrogate model and the target model have the same architecture [4, 5], mainly owing to the diversity of KGE model architectures. For instance, TransE utilizes linear projection on the features of triplets, while ConvE [10] introduces Convolutional Neural Networks (CNNs) and Deeppath [36] and PTransE [21] utilize Recurrent Neural Networks (RNNs). The poisoned triplets largely affect CNN and may fail to affect RNN since they learn different patterns. Therefore, prior efforts adopt the clear-box setting to study the vulnerability of KGE models against data poisoning attacks.

Furthermore, the relations between entities preserve strong semantical information. For instance, the relation *bornIn* should link an entity of the person type and an entity of the place type. In this work, we for the first time reveal that previous approaches neglect the semantic information because they only select triplets that maximize the attack goal, resulting in numerous ridiculous facts, e.g., (Obama, bornIn, Apple). These triplets could be easily detected by KG error detection methods, e.g., 80% poisoned triplets generated by [5, 39] are detected in our experiment. Considering that such error detection methods are often deployed before the training of KGE models [12, 15, 18], most poisoned triplets will be detected and removed before training which makes them easily defend against. For practice, to further study the vulnerability of KGE models against data poisoning attacks, we need to answer the following question: *Do there exist model-agnostic triplets that could infect different KGE models and satisfy semantic constraints?*

Our Work. We study the threat of data poisoning attacks on KGE models in more rigorous settings. We focus on the targeted attack on KGE - forcing the KGE model to predict specific wrong facts [5]. Take Fig. 1 as an example: the adversarial goal of the attacker is to force the prediction of - (*Despicable Me*, *Won*, *Oscar*). Once applied in downstream applications, some wrong decisions will be made. For instance, a corresponding knowledge graph-based Q&A system will answer *Oscar* when common users ask *which nomination does the movie Despicable Me won?* but give correct answers to unspecified questions. In summary, we aim to launch a data poisoning attack against KGE models that satisfies the following requirements:

- **Opaque-box Setting:** The effectiveness of the proposed data poisoning attack should be promised without the full knowledge of KGE models.
- **Semantic Constraints:** The inserted triplets should contain correct semantical information to evade the error detection methods of KG [15, 18, 30].
- **Stealthiness:** The infected model should maintain good performance on clean triplets [9, 40], preventing the undergoing attack from exposing by clean performance degradation.

To this end, we develop a framework satisfying the aforementioned constraints named model-agnostic semantic and stealthy (MaSS) data poisoning attack on KGE models. The key idea of MaSS is to insert indicative paths to mislead the target KGE model, e.g., inserting indicative paths composed of triplets *(Despicable Me, defeat, NomadLand)* and *(NomaLand, wonNomination, Oscar)* to make the model predict the fact *(Despicable Me, won, Oscar)*. The motivation behind the such design is that indicative paths on KGs represent how one entity is related to another semantically by some explicit relations and can be learned by various KGE models with different architectures [36]. Afterward, the remaining problems are: **(1)** how to search for indicative paths that represent the target fact? **(2)** how to force the triplets on the path to satisfy the semantic constraints? **(3)** how to make sure the poisoned triplets affect the target KGE model's prediction in an opaque setting?

For the first issue, we propose to translate the relation of the malicious fact to a sequence of relations, i.e., *path template*. For instance, we could translate *won* to *[defeat, wonNomination]*. To this end, we propose to utilize the path ranking algorithm (PRA) [19], which is able to extract representations of a relation by other relations on KGs, to conduct the path translation. For the second problem, we introduce the prior knowledge on the type constraints of relations [18], e.g., the entity that won Oscar could be a movie. Then we can determine the type of entities on the path. As such, the triplets on the path will satisfy the semantic constraints. Then the problem is how to determine the entities on the path. The main

challenge comes from how to pre-estimate the attack performance under the opaque-box setting. For this issue, we develop an opaque-box path injection theory, which shows the connection between opaque-box attack performance and the plausibility of triplets on the path. Specifically, the plausibility describes the logical consistency between a triplet and the whole KG [27], i.e., whether the triplet is more like a true fact on the KG. Hence, we propose an efficient algorithm to estimate the plausibility of triplets on the path and select optimal entities with a heuristic sampling strategy. Finally, since the injected triplets have high logical consistency to the clean KG, they will pose a little impact on normal utility, thus the stealthiness is also promised [27].

- We study a new opaque box and stealthy setting for the data poisoning attack on KGE and develop a framework called *MaSS* to generate effective triplets to bias different KGE models.
- We conduct theoretical analysis on the key factors determining the effectiveness and stealthiness of inserted triplets, and recognize the connection between the attack goal and plausibility of inserted triplets.
- We evaluate our attack on three benchmark KG datasets (FB15k-237, WN18RR, and CoDEx), and six representative KGE models demonstrating that the proposed attack achieves satisfied attack performance and stealthiness under the opaque box setting. Besides, none of the triplets injected by MaSS is detected by error detection methods.

2 RELATED WORK

Knowledge Graph Embeddings Knowledge graph representation methods aim at representing the facts on a knowledge graph to serve many knowledge acquisition tasks and various downstream applications [16, 31, 41]. Among numerous approaches, knowledge graph embedding (KGE) methods are growing rapidly to represent entities and relations in KG by converting them into a low dimensional vector space [6]. The vector-based representations make them be easily applied in various real-world tasks such as Q&A systems [16], recommendation systems [41] and neural translation methods [43], etc. KGE methods can be roughly categorized into three groups: translation based [6, 32], matrix factorization based [37] and deep learning based [10, 24]. For instance, the TransX models directly map the entities and relations to latent vectors, while deep learning based approaches add non-linear transformations on the embeddings to better capture the complex topological pattern such as the convolutional neural network used in the ConvKB [24]. The key design of KGE is to learn the indicative information of triplets, e.g., predicting the target entity given the source entity and the relation [21, 36]. As such, they could capture the indicative path during the KG reasoning [14]. Recently, to better model the reasoning path in KG, some work propose to explicitly model the reasoning path such as the DeepPath [36] and PTransE [21].

Data Poisoning Attacks on KGEs In the context of open-source KG, the attacker could submit malicious facts to manipulate the KG, leading to the biased learning of KGE models [4, 5, 26, 39]. Recent works propose to generate malicious facts by adversarial attack [26, 39], that is, crafting the triplets that maximize the adversarial goal. Generally, existing poisoning methods rely on a two-staged attack framework: (1) Search for embeddings that maximize the adversarial goal. (2) Select discrete triplets that have embeddings close to the adversarial embeddings. For instance, the work in [39] proposes to search the adversarial embeddings via the fast gradient sign method (FGSM) [13] and correspondingly select poisoned facts from a randomly sampled candidate set. [26] generates the embeddings by the influence function. Recent work [4, 5] utilizes relation inference to determine the optimal embeddings and inject paths to bias the prediction, which is similar to our work. However, as we have discussed, current methods adopt the white-box setting and neglect the semantic information of the injected triplets, which are important when we conduct the data poisoning attack on KG in a more practical setting.

Another line of related studies is data poisoning attacks against graph structure data such as the social networks [8], citation networks [22] and the recommender system [38]. Compared with the traditional graph structure, data poisoning attacks against triplet-formed KG is more difficult because: (1) KGs have heterogeneous structures but graph models require the nodes or links to be of only one type. (2) Lack of node attributes, e.g., in social networks, node attributes can be the age and gender information of user nodes.

3 PRELIMINARIES

3.1 Knowledge Graph Representations

Knowledge graph is often stored in **triplets**, e.g., *(source, relation, tail)*. Formally, we denote knowledge graph as a set of triplets $T = \{(s^{(i)}, r^{(i)}, t^{(i)})\}_{i=1}^{N}$, where $s^{(i)}, t^{(i)} \in E$ and $r^{(i)} \in R$ denote source and tail entities, and relations respectively. E, R denote the set of entities and relations from source to target entities respectively. Besides, triplets are often represented as graph structure data, where entities are nodes and relations are links, i.e., $\mathcal{G} = \{E, R, T\}$. Knowledge graph embeddings (KGE) model learns the low-dimensional representations $X \in \mathbb{R}^d$ of the entities $x_e \in X$ and the relations $x_r \in X$. The general objective of KGE is to preserve the structured relational information of KG by a scoring function g, which represents the plausibility for each triplet $(s, r, t) \in T$. The higher plausibility means that we could deduce the entity or the relation is given to the others in the triplet. An intensively used principle is the translation-based scoring function which represents the relations as translations from source to tail entities, e.g., TransE [6], $g(s, r, t) = -\|x_s + x_r - x_t\|_{1/2}$. Note that $\|\cdot\|_{1/2}$ is L_1 or L_2 norm. We denote the parameters of KGE models as θ, and the output of the scoring function is defined as $g_\theta(s, r, t)$.

Indicative paths. Indicative paths on the knowledge graph represent how one entity is semantically related to another by some relations, and are proved to be captured by KGEs [6]. The connective pattern plays an important role in KG reasoning. For instance, if there exists a path '$s \xrightarrow{\text{BornInCity}} e^{(1)} \xrightarrow{\text{CityInState}} e^{(2)} \xrightarrow{\text{StateInCountry}} t$', then it represents that s, t have the relationship of 'BornInCountry' because the relation path '⟨BornInCity, CityInState, StateInCountry⟩' have the same indicative information with relation 'bornIncountry'. Formally, a path is a sequence of connecting entities and relations on KG. We denote a L-length path from the entity s to the entity t on the knowledge graph $\mathcal{G}$ as $p_{s \to t}$, where each path is composed of an $(L+1)$-length entity path $\langle s, e_1, ..., e_{L-1}, t\rangle$ and an L-length relation path $\langle r_1, ..., r_L\rangle$. More specifically, the paths connecting two entities (s and t) may contain indicative information for the direct relationship between these two entities, e.g., 'BornInCountry'.

3.2 Threat Model

We first present the security settings. Knowledge graph often harvests data from open-source data, which opens an attack window for the adversaries to insert poisoned triplets. Especially, we assume the following threat model,

- **Access to the data**. The attacker has access to the knowledge graph $\mathcal{G}$ stores facts in the form of triplets T, e.g., the attacker generates and submits poisoned triplets to the open-source resources [2].
- **Opaque-box model**. Making the attack more practical, we assume that the adversary has no information about the victim KGE models. The only information that could be manipulated by the attacker is the KG
- **Attack constraints**. An attacker can only implement the attack by inserting a set of poisoned triplets, i.e., $\tilde{T}$, because it is hard to edit or remove existing facts in open-source KG by submitting poisoned triplets. Besides, we suggest a practical attack on the knowledge graph should additionally satisfy the following constraints - the adversary cannot create new entities or relations, cannot submit repetitive triplets, and is only allowed to submit a limited amount of poisoned triplets.

3.3 Problem Formulation

In this paper, we follow the previous work [39] and choose link prediction as the target task. This is because knowledge graphs are often incomplete, and link prediction is used to predict the missing relations between entities, which has been applied in various real-world applications including question answering [16, 31] and recommendation system [41]. Under the aforementioned threat settings, we use the attack goal similar to previous work [39], which aims to predict certain t^* given the query s^*, r^* as,

$$\max_{\tilde{T}} g_{\hat{\theta}}(s^*, r^*, t^*), \tag{1}$$
$$\text{s.t., } \hat{\theta} = \arg\max_{\theta} \sum_{(s,r,t)\in T\cup\tilde{T}} g_\theta(s, r, t).$$

where $\tilde{T}$ is the set of poisoned triplets. To accomplish the attack, the attacker injects $\tilde{T}$ into clean KG. Besides, we consider the data poisoning attacks that satisfy the requirements of opaque setting, semantic constraints, and stealthiness.

4 METHODOLOGY

In this section, we present a data poisoning attack against KGE models that are model-agnostic, semantic, and stealthy named *MaSS*.

4.1 Opaque-box Path Injection Theory

4.1.1 Indicative Path Injection. To force various KGE models to predict the target fact, we focus on the *indicative path*. Specifically, although the model structures are different, the connective patterns in paths can be well learned by different KGE models in related literature [36]. For instance, to force the prediction of *(Bob, bornIn, USA)*, we could insert triplets *(Bob, bornIn, NewYork)* and (*NewYork, locateIn, USA*).Then we rewrite the attack goal in Eq. 1 by,

$$\max_{\tilde{T}} g_{\hat{\theta}}(s^*, r^*, t^*), \tag{2}$$
$$\text{s.t., } \theta^* = \arg\max_{\theta} \sum_{(s,r,t)\in T\cup\tilde{T}} g_\theta(s, r, t), \quad \tilde{T} = \{p^{(j)}_{s^*\to t^*}\}_{j=1}^{n}$$

where $p^{(j)}_{s^*\to t^*}$ represents a path from s^* to t^*, i.e., the *indicative path*, and for each target fact we inject n indicative paths. The representation of the path can be described as,

$$p_{s^*\to t^*} = s^* \xrightarrow{\tilde{r}_1} \tilde{e}_1 \cdots \xrightarrow{\tilde{r}_{L-1}} \tilde{e}_{L-1} \xrightarrow{\tilde{r}_L} t^*. \tag{3}$$

where L is the length of the path. Then the remaining problem is, how to pre-estimate the attack effectiveness of an indicative path. An intuitive approach is to measure the influence of each candidate path by injecting the paths into KGs and then training the KGE models. However, this strategy is not only time-consuming but also impractical under the opaque-box setting. To address the issue, we first theoretically pre-estimate the attack effectiveness under the clear-box setting, then discuss how to transfer the estimated attack effectiveness to the opaque-box setting.

4.1.2 Pre-estimation of Attack Effectiveness. First, we present Theorem 1 proving that when attacking TransE [6] in clear-box setting.

Theorem 1. *Suppose there are n paths - $\{p^{(j)}_{s^*\to t^*}\}_{j=1}^{n} = \{\{s^*, \tilde{r}_1^{(j)}, \tilde{e}^{(j)}, \tilde{r}_2^{(j)}, t^*\}\}_{j=1}^{n}$ with an entity pair (s^*, t^*) and corresponding KGE model TransE with parameter θ. The lower bound of the attack effectiveness is,*

$$g_\theta(s^*, r^*, t^*) \geq \max_{1\leq j\leq n} (-\|x_{r^*} - x_{\tilde{r}_1^{(j)}} - x_{\tilde{r}_2^{(j)}}\|_2 + \delta_1^{(j)} + \delta_2^{(j)}). \tag{4}$$

where $\delta_1^{(j)} = g_\theta(s^*, \tilde{r}_1^{(j)}, \tilde{e}^{(j)})$, $\delta_2^{(j)} = g_\theta(\tilde{e}^{(j)}, \tilde{r}_2^{(j)}, t^*)$, and $x_\cdot$ is the embedding of TransE.

The omitted proof is presented in Appendix. Note that the results could be also extended to paths with $L \geq 2$.

After this, we present Theorem 2 proving that when attacking an unknown KGE model, the attack performance in opaque-box setting of poisoned triplets in Theorem 1 do not degrade compared with the results from the TransE model with a certain probability.

Theorem 2. *Suppose that TransE's score of a given triplet is sampled from a gaussian distribution $g_\theta \sim \mathcal{N}(\mu_0, \sigma_0^2)$ and another KGE model's score is sampled from $g_e \sim \mathcal{N}(\mu_1, \sigma_1^2)$. If g_e learns triplet more accurately than TransE, i.e., $\mu_1 > \mu_0$. Then if the following condition is held, i.e., $\sigma_1 < \sqrt{\frac{(\mu_0-\mu_1)^2}{\Phi^{-1}(1-\delta)^2} - \sigma_0^2}$ where $\Phi(\cdot)$ is the cumulative distribution function of normal distribution. The score g_e will be larger than g_θ with the probability of δ and $\delta > \frac{1}{2}$ for any triplet.*

The omitted proof is presented in Appendix.

4.1.3 Summary. The proposed opaque-box path injection theory mainly states the following results: (1) Injecting indicative paths could make various KGE models predict the target fact; (2) The attack effectiveness of indicative paths under the clear-box setting (i.e., the TransE model) is determined by two factors: the similarity between the relation path $\langle r_1, r_2\rangle$ and the target relation r^*, and The plausibility of triplets on the path; (3) The attack effectiveness of indicative paths under the opaque-box setting does not degrade with a certain probability when we estimate it by a TransE model.

4.2 Path Translation

To translate the target relation to a sequence of relations, i.e., finding $\langle r_1, r_2\rangle$ that is the most similar to the target relation r^*, we introduce the technique of path ranking algorithm (PRA), which is an effective

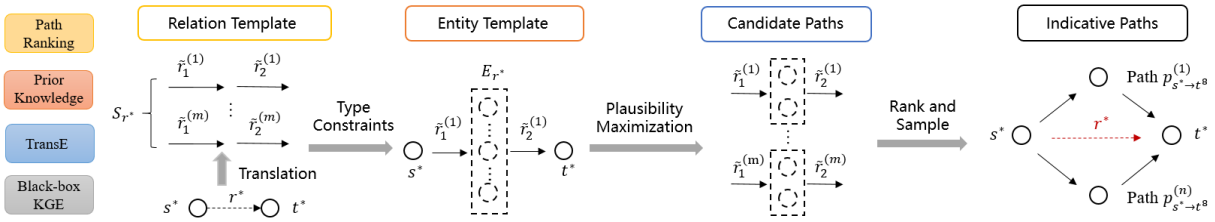

Figure 2: Overview of the proposed MaSS on KGE models.

tool for relation translation in related literature [19, 20]. The goal of PRA is to learn whether there exists a similarity between $\langle r_1, r_2\rangle$ and r for any $r_1, r_2, r \in R$. To this end, the PRA leverages a supervised classification model to learn the similarity. Specifically, for a relation path $\langle r_1, r_2\rangle$, if there exist two entities not only connected by the path but also have direct relation r_+, then r_+ is regarded as the positive label for $\langle r_1, r_2\rangle$. On the other side, we could randomly sample a relation r_- as the negative label. Finally, we generate numerous training pairs and learn a classification model.

After learning PRA, we could translate the target relation r^* by selecting $\langle r_1, r_2\rangle$ that are predicted as r^*. Specifically, for all relation pairs $R \times R$, if the confidence score of prediction on r^* is larger than 0.9, we regard it as the translation of the target relation r^*. As such, given the target fact (s^*, r^*, t^*), we could generate a candidate set of translated relation path S_{r^*}. Then the indicative path for the target fact could be represented by,

$$p_{s^*\to t^*} = s^* \xrightarrow{\tilde{r}_1} \tilde{e} \xrightarrow{\tilde{r}_2} t^*, \quad s.t. \ \langle \tilde{r}_1, \tilde{r}_2\rangle \in S_{r^*} \tag{5}$$

Note that there often exist multiple potential $\langle \tilde{r}_1, \tilde{r}_2\rangle$ for the relation r^*, i.e., the *path template* $|S_{r^*}| \geq 1$.

4.3 Semantic Constraints

Given extracted path templates, we need to determine the entities of triplets on the paths - $(s^*, \tilde{r}_1, \tilde{e})$ and $(\tilde{e}, \tilde{r}_2, t^*)$. To satisfy the semantic requirements, prior knowledge of relation constraints constrains the type of entities connected by a certain relation. For instance, a relation 'BornInCity' should connect an entity of a person and an entity of a city [18]. Such prior knowledge could be obtained by the rich ontology [18], which can be found in various open-source KG such as *Dbpedia* [2]. Formally, given an arbitrary relation r, we denote the restricted type of source entity as $domain[r]$. Similarly, the restricted type of tail entity is represented as $range[r]$. Then the indicative path $p_{s^*\to t^*}$ could be reformulated as,

$$\begin{aligned} &p_{s^*\to t^*} = s^* \xrightarrow{\tilde{r}_1} \tilde{e} \xrightarrow{\tilde{r}_2} t^* \\ &s.t. \langle \tilde{r}_1, \tilde{r}_2\rangle \in S_{r^*}, \quad \tilde{e} \in E_{range[\tilde{r}_1]} \cap E_{domain[\tilde{r}_2]}. \end{aligned} \tag{6}$$

where $E_{range[\cdot]}$ and $E_{domain[\cdot]}$ are the sets of entities whose type are $range[\cdot]$ and $domain[\cdot]$ respectively. Then given a target malicious fact (s^*, r^*, t^*), the path template is represented by the relation template S_{r^*} and entity template $E_{r^*} = \{E_{range[\tilde{r}_1]} \cap E_{domain[\tilde{r}_2]}, \forall \langle \tilde{r}_1, \tilde{r}_2\rangle \in S_{r^*}\}$.

4.4 Path Generation

4.4.1 Entity Selection. Then we introduce how to determine the entities that not only satisfy the above constraints but also maximize the plausibility of triplets on the path under the clear-box setting. The main challenge comes from the huge combinational search space, that is, we should enumerate all entities that satisfy the semantic constraints. To address the issue, we develop a heuristic algorithm to search for solutions. The key design of the proposed method is similar to the beam search in related literature [35]. Specifically, given malicious fact (s^*, r^*, t^*), the strategy of selection is as follows: **(i)** Generate the path template S_{r^*} and E_{r^*}. **(ii)** Select the top-m relation paths in S_{r^*} by maximizing $-\|x_{r^*} - x_{\tilde{r}_1} - x_{\tilde{r}_2}\|_{1/2}$. **(iii)** Select top-$\tilde{n}$ entities $\tilde{e}$ of each relation path which maximizes $g_\theta(s^*, \tilde{r}_1, \tilde{e})$ and $g_\theta(\tilde{e}, \tilde{r}_2, t^*)$. **(iiii)** Aggregate $m \times \tilde{n}$ paths and select top-n paths that maximize the lower bound.

Compared with the original greedy search, the candidate set could largely reduce the search space, besides, it could also alleviate the problem of sub-optimal solutions caused by greedy search.

4.4.2 Random Sampling Strategy. Finally, we present how to tackle the opaque-box setting given the paths generated by the above strategy. As Theorem 2 demonstrates, if the triplets on the path are easy to be learned by TransE, they will also have larger plausibility in other models since TransE could be regarded as the basic KGE model. However, the larger plausibility is not always correct, instead, *it holds with large probability,* $1 - \delta$. This explains why some triplets have large plausibility on TransE but smaller ones on other models, although TransE is one of the basic KGE models. If we purely maximize the lower bound based on TransE, the injected paths may not work on other models due to overfitting. To address the issue, we improve the aggregation mechanism by random sampling. Specifically, after we generate $m \times \tilde{n}$ paths by the path templates and plausibility maximization, we sort the $m \times \tilde{n}$ by the lower bound. Then instead of directly obtaining top-n paths, we enumerate the paths in sequence and accept them with probability $1 - \delta$. Such a mechanism will help the generated triplets not be overfitted for the surrogate model.

4.4.3 Stealthiness. One advantage of our attack is that the poisoned triplets have large plausibility, which can hardly impact the model learning on clean data. This is similar to the stealthiness objective of other data poisoning attacks [42]. Furthermore, inspired by previous studies on influence propagation on traditional graph structure [23], we propose to control the degrees of the selected entities to enhance stealthiness. That is, entities with high degrees are more likely to propagate their negative influence to other entities than those with low degrees. Therefore, we select the entities with lower degrees so that their negative impacts are limited.

4.4.4 Discussion. The detailed algorithm is shown in Alg. 1. We present the time complexity analysis of the proposed attack framework. The search space of optimizing Eq. 1 is $O(|E|^2|R|)$ for a given malicious fact, making it hard to implement on real-world datasets. Our proposed algorithm substantially reduces the computational complexity. As shown in Algorithm 1, the computational complexity is much less than $O(|E| * m)$ for each malicious triplet, which is

dominant by the intermediate entity selection in Line four. Hyper-parameter m is often set small (e.g., $m = 10$), thus the attack strategy scales linearly with the number of entities in the target KG.

5 EMPIRICAL EVALUATION

We evaluate our proposed attack on three benchmark datasets, attacking six state-of-the-art KGE models. Specifically, our experiments are designed to answer the following research questions: **RQ1** - Can MaSS successfully attack in opaque-box settings? **RQ2** - Can MaSS conduct a stealthy attack? **RQ3** - Do the injected triplets of MaSS contain semantical information?**RQ4** - How do settings influence performance and stealthiness?

5.1 Experiment Setups

Datasets Three real-world knowledge graphs are used in our experiments: *FB15k-237* [33], *WN18RR* [6] and *CoDEx* [28]. Note that there are three sizes of CoDEx and we choose CoDEx-M in this paper. The detailed statistics of datasets are summarized in Table. 4. The validation set is used to help determine the hyper-parameters of KGE models.

Besides, to evaluate the effectiveness and stealthiness, we prepare two kinds of test sets based on original test data: clean test data (i.e., *Set A*), to evaluate the stealthiness; target triplets (i.e., *Set B*=$\hat{T}$), to validate the attack effectiveness. Specifically, *Set A* is the original test set and is used to validate the attack stealthiness. For *Set B*, we first randomly sample 500 triplets from the original test set and then replace the original entity with attacker-target one t^* with low plausibility. Specifically, we pre-train a TransE [6] and a ConvKB [24] and select t^* with the predicted rank around 500. The reported results are the averaged value of 10 independent attacks.

Victim Models. For the victim models, we cover 6 state-of-the-art KGE models: translation distance model: *TransE* [6] and *TransH* [34]; tensor decomposition model: *DisMult* [37]; deep learning based model: *ConvKB* [24] and *ConvE* [10]; and path-enhanced model: *Deeppath* [36]. For the surrogate model, we leverage TransE trained on the benign KG. For the implementation of victim and surrogate models, we rigorously follow their authors' suggestions to set hyper-parameters, except for the embedding dimensions which are both fixed to 128 for FB15k-237, WN18RR and CoDEx respectively on each model. After generation, they are injected into the clean datasets, and then the victim models are retrained from scratch on the poisoned dataset until convergence.

Baselines & Implementation Details. We compare our proposed attack with two state-of-the-art poisoning attacks on knowledge graph embedding: *DirectAdd* [39] and *ComAttack* [5]. To conduct a model-agnostic attack, we use the same surrogate model as our attack used to generate plausibly malicious triplets for all the victim models. The number of malicious triplets of both baselines and proposed attack, e.g., n, is set as 2. The number of selected relation paths, e.g., m is set as 10. The number of the selected entities, e.g., $\tilde{n}$ is set as 10. In random sampling, δ is set as 0.7.

Evaluation Metrics We follow the evaluation protocol in common link prediction task [6]. Given a test sample (s, r, t), we first replace the tail entity by entity from E, and use KGE models to predict the scores of all entities. The scores are then sorted in descending order and the rank of the correct entity is determined. We choose Hit@10 (Hits at 10, if the rank of the correct entity is lower than 10) and MRR (Mean Reciprocal Rank) as the performance metrics [6]. Besides, we calculate Hit@10 and MRR on tail entities in *Set B* predicted by a poisoned model to measure attack performance. *Higher Hit@10 and MRR mean better attack effectiveness.* We calculate Hit@10 and MRR on common test triplets in *Set A* after attack for stealthiness. *Higher Hit@10 and MRR imply better attack stealthiness.*

5.2 Attack Effectiveness (RQ1)

5.2.1 Clear-box attack performance. As the column of 'TransE' in Table 1 shows, our proposed attack strategy substantially outperforms the baseline DirectAdd and ComAttack under the clear-box setting. Hit@10 of two benchmark datasets exceed 0.8, and this demonstrates that 80% of tail entity will be predicted in the top-10 of the ranking list. On the CoDEx dataset, the more robust dataset, Hit@10 aslo exceeds 0.5. That is, the proposed attack is significant on both benchmark datasets and robust datasets. As for two baselines, Hit@10 are both lower than 0.6 on two benchmark datasets, e.g., the highest is 0.56 of ComAttack attacking WN18RR, and the lowest is 0.07 of DirectAdd attacking CoDEx. As for metric MRR, the proposed attack achieves $0.39 \sim 0.62$, which demonstrates that our attack can efficiently improve the ranks of half-target triplets posing severe threats to knowledge graph embedding. What's more, we also investigate the attack performance and stealthiness with different attack budgets in Fig. 3 and Fig. 4. Our proposed attack is the only one that can balance the attack effectiveness and stealthiness.

The attack performance is different among the three datasets. All attacks are more significant on FB15k-237 than on WN18RR. The main reason for the difference is that WN18RR is sparser than FB15k-237, making the negative influence of poisons harder to propagate to targets and on the whole graph. Second, the stealthiness of all attacks is more significant on FB15k-237 than on WN18RR. This is because the average number of triplets that each entity involves in WN18RR is significantly smaller than that in FB15K-237. Hence, the graph structure of FB15K-237 is more stable and robust. Studies like [39] have the same observations as ours. The attack performance of all three attack strategies is less significant on CoDEx.

5.2.2 Attack in opaque setting. Furthermore, we focus on investigating attacks under the opaque-box setting. As Table 1 shows, our proposed attack also exhibits attack effectiveness under opaque-box settings. The attack effectiveness (Hit@10 and MRR) is not going to drop except ConvE on three datasets. The drops do not surpass 0.15 on Hit@10 and 0.03 on MRR. We infer the main reason is that ConvE has a very different model architecture compared with TransE and the model is easier to overfit. Besides, our proposed attack can achieve 100% Hit@10 and up to 0.5 MRR against Deeppath. This demonstrates the efficiency of the proposed path injection theory [36]. DirectAdd and ComAttack are not stable under opaque-box setting. For DirectAdd, once transferred to attack models that are different from TransE, the attack performance is going to degrade. For instance, when attacking Deeppath [36] using RNNs or ConvKB [24] using CNNs, both Hit@10 and MRR are near 0. Furthermore, ComAttack also proposes to inject relation paths and can somewhat develop transferability. However, they still rely on embeddings of the surrogate model and show poor transferability when attacking models like DisMult and ConvE. The attack performance drops

Table 1: Attack performance on different infected models. The column with $^+$ represents attack performance under the clear-box setting. (* the best result and _ the important results)

		TransE$^+$		TransH		DisMult		ConvE		ConvKB		Deeppath	
		Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR
FB15k-237	DirectAdd	0.48	0.15	0.29	0.07	0.33	0.08	0.11	0.05	0.18	0.06	0.31	0.08
	ComAttack	0.27	0.10	0.22	0.08	0.08	0.01	0.10	0.04	0.11	0.04	0.27	0.07
	Ours	0.8*	0.39*	0.8*	0.40	0.8*	0.38*	0.55	0.23*	0.82*	0.38*	1*	0.62*
WN18RR	DirectAdd	0.36	0.14	0.12	0.06	0.17	0.11	0.21	0.09	0.0	0.00	0.00	0.01
	ComAttack	0.56	0.21	0.26	0.08	0.44	0.19	0.42	0.19	0.54	0.22	0.76	0.3
	Ours	0.8*	0.45*	0.62*	0.30*	0.77*	0.43*	0.96*	0.41*	0.8*	0.39*	1*	0.55*
CoDEx	DirectAdd	0.07	0.02	0.05	0.02	0.43	0.21	0.22	0.10	0.21	0.11	0.09	0.02
	ComAttack	0.15	0.05	0.11	0.03	0.42	0.21	0.24	0.22	0.22	0.11	0.16	0.06
	Ours	0.57*	0.28*	0.33*	0.13*	0.54*	0.26*	0.40*	0.26*	0.41*	0.17*	0.49*	0.24*

Table 2: Clean performance (stealthiness) against different attacks. The column with $^+$ represents clean performance under clear-box setting. (* the best result and _ the important results)

		TransE$^+$		TransH		DisMult		ConvE		ConvKB		Deeppath	
		Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR
FB15k-237	Clean	0.42	0.29	0.40	0.28	0.42	0.24	0.42	0.30	0.50	0.32	0.59	0.35
	DirectAdd	0.42*	0.25	0.38	0.23	0.35	0.032	0.45*	0.29	0.40	0.23	0.42	0.24
	ComAttack	0.42*	0.26	0.41	0.23	0.37	0.23	0.41	0.23	0.42	0.25*	0.42	0.24
	Ours	0.42*	0.30*	0.43*	0.26*	0.42*	0.41*	0.44	0.31*	0.43*	0.23	0.48*	0.29*
WN18RR	Clean	0.51	0.26	0.54	0.28	0.39	0.23	0.52	0.40	0.52	0.24	0.55	0.33
	DirectAdd	0.40	0.14	0.34	0.12	0.36	0.18	0.44	0.23	0.43	0.29	0.52*	0.19
	ComAttack	0.49	0.19*	0.36	0.13	0.37*	0.18	0.38	0.23	0.40	0.24	0.46	0.21
	Ours	0.50*	0.18	0.48*	0.20*	0.36	0.26*	0.51*	0.31*	0.48*	0.35*	0.47	0.22*
CoDEx	Clean	0.52	0.24	0.53	0.24	0.51	0.22	0.52	0.25	0.52	0.24	0.54	0.28
	DirectAdd	0.51*	0.22*	0.51*	0.23	0.43	0.23	0.22	0.10	0.22	0.11	0.52*	0.23*
	ComAttack	0.51*	0.22*	0.51*	0.24*	0.43	0.23	0.23	0.11	0.24	0.11	0.52*	0.23*
	Ours	0.51*	0.21	0.51*	0.23	0.49*	0.26*	0.49*	0.24*	0.50*	0.23*	0.52*	0.23*

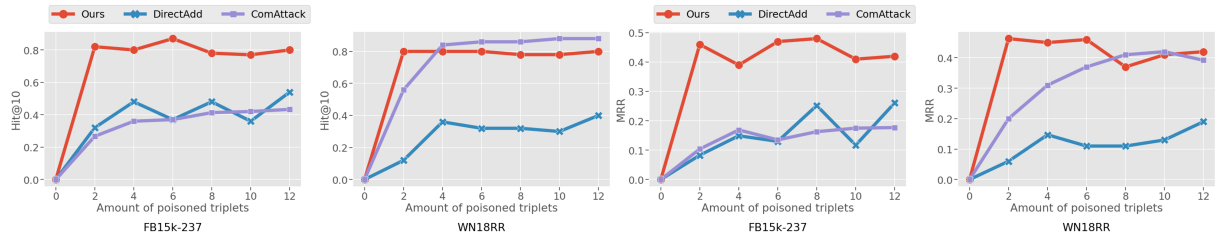

Figure 3: Attack performance (Hit@10 and MRR) w.r.t. amount of poisoned triplets. The metrics are calculated on Set B.

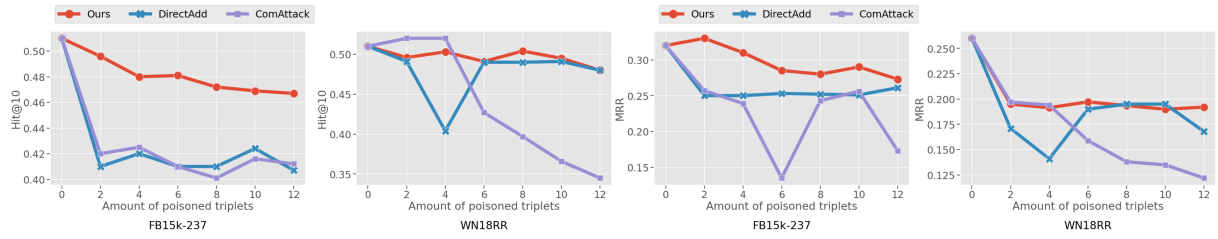

Figure 4: Attack stealthiness (Hit@10 and MRR) w.r.t. amount of poisoned triplets. The metrics are calculated on Set A.

by the rate of 50% on FB15k-237, 20% – 30% on WN18RR. What's more, baseline attacks achieve higher attack performance against DisMult, ConvKB, and ConvE on the CoDEx dataset. According to our results, we find that DisMult is more vulnerable than TransE. DisMult is a matrix factorization-based method and a similar problem of matrix factorization has been investigated in many other areas [11]. Although the attack performance on ConvE and ConvKB are significant, the clean performance of these two models is very low. Besides, we observe that when attacking opaque-box models, DirectAdd and ComAttack sometimes maintain a higher clean performance on FB15k-237 and WN18RR, but the attack performance of these two attacks is not very significant.

5.3 Attack Stealthiness (RQ2)

Moreover, to achieve substantially improved attack effectiveness, our proposed attack does not sacrifice the normal utility of the

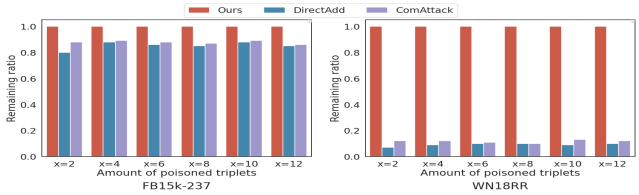

Figure 5: Remaining ratio of poisoned triplets filtered by type-constraints [18].

victim model as shown in Table 2. As we can see from the results, both baseline models will negatively impact the performance of learning clean triplets with Hit@10 dropping by 10% – 20% on average. Especially, when attacking ConvE and ConvKB, the clean performance is too low to make an accurate prediction. And this may render the presence of the attack detectable. The clean performance of our attack under clear-box setting does not drop, but the clean performance under opaque-box setting may drop by 3% – 5%.

5.4 Detection of Poisoned Triplets (RQ3)

We verify whether the poisoned triplets contain correct semantic information. We leverage the prior knowledge of type constraint to detect source and tail entities of injected triplets. Fig. 5 shows the detection results. We can see from the figure, none of the triplets generated by our proposed attack will be detected. For the other two datasets, almost 80% of generated triplets against WN18RR will be detected and 20% against FB15k-237 will be detected. Furthermore, we adopt an approach to detect mutually exclusive facts by using the training set to filter the injected facts, i.e., Fig. 7. If a poisoned fact shares the same relation and tail or source and relation with existing facts, this fact will not be allowed to inject.

5.5 In-depth Analysis of MaSS (RQ4)

Hyper-parameter sensitivity Afterward, we also investigate the hyper-parameter of the surrogate TransE model. The most important parameter of TransE is the embedding size. In Fig. 9, the appropriate embedding size is 128 for both datasets, and a larger embedding size achieves better attack performance.

Time efficiency of different attack strategies are shown in Fig. 6. We find that DirectAdd is more time efficient than our attack and ComAttack [5]. This is because DirectAdd leverages the fast sign gradient method (FGSM) [13] to generate poisoned triplets which is known to be efficient [13]. The search space of our attack is largely reduced. Furthermore, the denser the knowledge graph, the more inefficient ComAttack is. For instance, FB15k-237 is denser than WN18RR, and time of poisoning FB15k-237 is much longer.

Enhancing existing attacks We leverage type constraint and plausibility constraint to enhance two baseline attacks. The results are shown in Table 3. We find that: (i) Type constraints largely harm the attack performance of DirectAdd because DirectAdd relies on injecting abnormal triplets to largely bias the behavior of KGE. But triplets satisfying the type constraint are not able to maximize the attack goal of DirectAdd. (ii) With large plausibility constraints, the stealthiness of both attacks is enhanced.

Semantic information we investigate if the plausibility maximization in our proposed attack can result in large plausibility of

Table 3: Utilizing optimizations of the proposed attacks to enhance existing attacks - DirectAdd and ComAttack

	Type constraint				Plausibility constraint			
	Before enhanced		After enhanced		Before enhanced		After enhanced	
	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR
DirectAdd	0.48	0.15	0.21	0.18	0.42	0.29	0.44	0.30
ComAttack	0.27	0.10	0.29	0.11	0.42	0.26	0.43	0.27

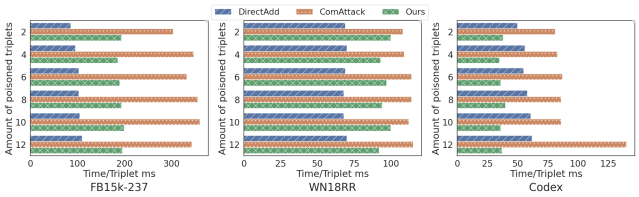

Figure 6: Time per attacked triplet on three datasets.

poisoned triplets, i.e., Fig. 8. The plausibility is calculated by ConvE trained on the poisoned dataset (FB15k-237). The confidence of generated poisoned triplets is similar to clean data while larger than noisy data. We present several poisoned triplets generated by the proposed attack in Table 5.

Potential defense. As the proposed attack can not be detected by existing error detection methods, We suggest defending against such attacks from the system level, e.g., identity authentication of users and protecting the triplet database from malicious tampering.

6 CONCLUSION

In this paper, we consider a more realistic setting for data poisoning attacks on KGEs in the opaque-box setting, and the attack stealthiness and the semantic correctness of poisoned data. To this end, we develop a novel attack framework based on path injection. Our study points out a new direction for data poisoning attacks on knowledge graphs. As for the future study, although we have considered a more realistic setting for a data poisoning attacks on KGE, to conduct the attack in the physical world, we need to submit text-based facts to the open-source KG instead of triplets. Such an attack is extremely difficult to implement since we need to generate a sentence or document given a poisoned triplet. Furthermore, it is also interesting to observe the performance in downstream applications such as the recommender system. We will consider these points and leave the attack against the commercial systems as a future study to validate the impact of our work.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program (2021YFB3101200), National Natural Science Foundation of China (61972099, U1736208, U1836210, U1836213, 62172104, 62172105, 61902374, 62102093, 62102091). Min Yang is a faculty of Shanghai Institute of Intelligent Electronics & Systems, Shanghai Collaborative Innovation Center of Intelligent Visual Computing and Engineering Research Center of Cyber Security Auditing and Monitoring, Ministry of Education, China. Mi Zhang and Min Yang are the corresponding authors.

REFERENCES

[1] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, M. Harris, D. Hill, L. Issel-Tarver, A. Kasarskis, Suzanna Lewis, J. C. Matese, J. Richardson, M. Ringwald, G. Rubin, and G. Sherlock. 2000. Gene Ontology: tool for the unification of biology. *Nature Genetics* 25 (2000), 25–29.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *ISWC*.

[3] Prithu Banerjee, Lingyang Chu, Yong Zhang, Laks V. S. Lakshmanan, and Lanjun Wang. 2021. Stealthy Targeted Data Poisoning Attack on Knowledge Graphs. In *ICDE*.

[4] Peru Bhardwaj, John D. Kelleher, Luca Costabello, and Declan O'Sullivan. 2021. Adversarial Attacks on Knowledge Graph Embeddings via Instance Attribution Methods. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 8225–8239. https://doi.org/10.18653/v1/2021.emnlp-main.648

[5] Peru Bhardwaj, John D. Kelleher, Luca Costabello, and Declan O'Sullivan. 2021. Poisoning Knowledge Graph Embeddings via Relation Inference Patterns. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 1875–1888. https://doi.org/10.18653/v1/2021.acl-long.147

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[7] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2021. Dual Quaternion Knowledge Graph Embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6894–6902.

[8] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial Attack on Graph Structured Data. In *ICML (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 1123–1132. http://proceedings.mlr.press/v80/dai18b.html

[9] Kemal Davaslioglu and Yalin E Sagduyu. 2019. Trojan attacks on wireless signal classification with adversarial machine learning. In *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 1–6.

[10] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*.

[11] Wenqi Fan, Tyler Derr, Xiangyu Zhao, Yao Ma, Hui Liu, Jianping Wang, Jiliang Tang, and Qing Li. 2021. Attacking Black-box Recommendations via Copying Cross-domain User Profiles. In *ICDE*.

[12] R. Fasoulis, Konstantinos Bougiatiotis, Fotis Aisopos, Anastasios Nentidis, and Georgios Paliouras. 2020. Error detection in Knowledge Graphs: Path Ranking, Embeddings or both? *CoRR* abs/2002.08762 (2020). arXiv:2002.08762 https://arxiv.org/abs/2002.08762

[13] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6572

[14] William L. Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. 2018. Embedding Logical Queries on Knowledge Graphs. In *Advances in Neural Information Processing Systems*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 2030–2041.

[15] Yan Hong, Chenyang Bu, and Tingting Jiang. 2020. Rule-enhanced Noisy Knowledge Graph Embedding via Low-quality Error Detection. In *2020 IEEE International Conference on Knowledge Graph, ICKG 2020, Online, August 9-11, 2020*, Enhong Chen and Grigoris Antoniou (Eds.). IEEE, 544–551. https://doi.org/10.1109/ICBK50248.2020.00082

[16] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge Graph Embedding Based Question Answering. In *WSDM*, J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman (Eds.). ACM, 105–113. https://doi.org/10.1145/3289600.3290956

[17] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *WSDM*.

[18] Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-constrained representation learning in knowledge graphs. In *International semantic web conference*. Springer, 640–655.

[19] Ni Lao, Tom M. Mitchell, and William W. Cohen. 2011. Random Walk Inference and Learning in A Large Scale Knowledge Base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 529–539. https://aclanthology.org/D11-1049/

[20] Ni Lao, Jun Zhu, Xinwang Liu, Yandong Liu, and William W. Cohen. 2010. Efficient Relational Learning with Hidden Variable Detection. In *Advances in Neural Information Processing Systems*, John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta (Eds.). 1234–1242.

[21] Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling Relation Paths for Representation Learning of Knowledge Bases. In *EMNLP*, Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton (Eds.).

[22] Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. 2022. Adversarial Attack on Graph Neural Networks as An Influence Maximization Problem. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang (Eds.). ACM, 675–685. https://doi.org/10.1145/3488560.3498497

[23] Jiaqi Ma, Shuangrui Ding, and Qiaozhu Mei. 2020. Towards More Practical Adversarial Attacks on Graph Neural Networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).

[24] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2017. A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121* (2017).

[25] Heiko Paulheim and Aldo Gangemi. 2015. Serving DBpedia with DOLCE–more than just adding a cherry on top. In *International semantic web conference*. Springer, 180–196.

[26] Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. 2019. Investigating robustness and interpretability of link prediction via adversarial modifications. *arXiv:1905.00563* (2019).

[27] Meng Qu, Junkun Chen, Louis-Pascal A. C. Xhonneux, Yoshua Bengio, and Jian Tang. 2021. RNNLogic: Learning Logic Rules for Reasoning on Knowledge Graphs. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=tGZu6DlbreV

[28] Tara Safavi and Danai Koutra. 2020. Codex: A comprehensive knowledge graph completion benchmark. *arXiv preprint arXiv:2009.07810* (2020).

[29] Amit Singhal. 2012. Google Knowledge Graph. (2012).

[30] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. 2019. Label-Consistent Backdoor Attacks. *CoRR* abs/1912.02771 (2019). arXiv:1912.02771 http://arxiv.org/abs/1912.02771

[31] Min Wang, Yanzhen Zou, Yingkui Cao, and Bing Xie. 2019. Searching software knowledge graph with question. In *International Conference on Software and Systems Reuse*. Springer, 115–131.

[32] 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 (2017), 2724–2743.

[33] Rui Wang, Bicheng Li, Shengwei Hu, Wenqian Du, and Min Zhang. 2019. Knowledge graph embedding via graph attenuated attention networks. *IEEE Access* 8 (2019), 5212–5224.

[34] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. *AAAI*.

[35] Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. 2021. Graph backdoor. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*.

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

[37] 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*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6575

[38] Hengtong Zhang, Changxin Tian, Yaliang Li, Lu Su, Nan Yang, Wayne Xin Zhao, and Jing Gao. 2021. Data Poisoning Attack against Recommender System Using Incomplete and Perturbed Data. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 2154–2164. https://doi.org/10.1145/3447548.3467233

[39] Hengtong Zhang, Tianhang Zheng, Jing Gao, Chenglin Miao, Lu Su, Yaliang Li, and Kui Ren. 2019. Data poisoning attack against knowledge graph embedding. *arXiv:1904.12052* (2019).

[40] Xinyang Zhang, Zheng Zhang, and Tianying Wang. 2021. Trojaning Language Models for Fun and Profit. *EuroS&P* (2021).

[41] Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192*.

[42] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. 2020. Clean-Label Backdoor Attacks on Video Recognition Models. In *CVPR*.

[43] Yang Zhao, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2020. Knowledge Graphs Enhanced Neural Machine Translation. In *IJCAI*.

Algorithm 1 Generating poisoned triplets.

Input: Target triplet (s^*, r^*, t^*), pre-trained PRA and the surrogate model g_θ, amount of injected path n, relation template and type constraints S_{r^*}, E_{r^*}.
Output: Poisoned triplet set $\tilde{T}$.
$\tilde{T} = \emptyset$
Select m relation paths from S_{r^*} with the largest $-\|x_{r^*} - x_{\tilde{r}_1} - x_{\tilde{r}_2}\|_2$ by surrogate model g_θ
for each selected relation path $\langle \tilde{r}_1, \tilde{r}_2 \rangle$ **do**
  Select $\tilde{n}$ entities $\tilde{e}$ from E_{r^*} with largest $\delta_1 + \delta_2$, and add $(\tilde{r}_1, \tilde{e}, \tilde{r}_2)$ to the candidate set $\tilde{P}_{s^* \to t^*}$.
end for
Select n paths $p_{s^* \to t^*}$ from $\tilde{P}_{s^* \to t^*}$ that maximize score in Theorem 1, and add them to $\tilde{T}$.
return $\tilde{T}$

Table 4: Statistics of three datasets.

	FB15k-237	WN18RR	CoDEx
# Entities (E)	14,541	40,943	17,050
# Relations (R)	237	11	51
# Train Set (T)	272,115	86,835	185,584
# Valid Set	17,535	3,034	10,310
# Set A/B ($\hat{T}$)	19,966/500	2,634/500	9,000/500

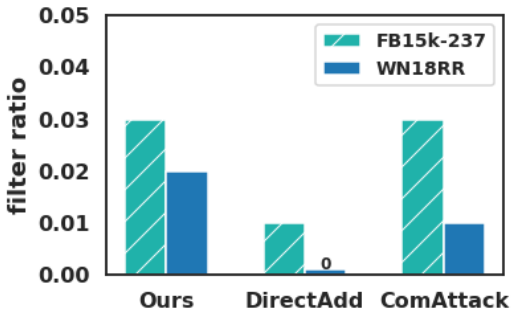

Figure 7: Mutual exclusive fact detection of baseline attacks and our attack.

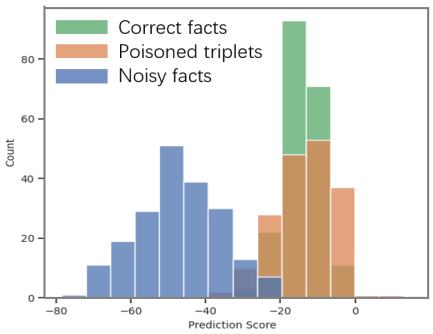

Figure 8: Plausibility of noisy facts, poisoned triplets and correct facts.

A APPENDIX

A.1 Technical Proof

Proof of Theorem 1.

Proof. According to TransE's [6] score function on triplets, we have,

$$g_\theta(s^*, \tilde{r}_1, \tilde{e}) = -\|x_{s^*} + x_{\tilde{r}_1} - x_{\tilde{e}}\|_{1/2},$$
$$g_\theta(\tilde{e}, \tilde{r}_2, t^*) = -\|x_{\tilde{e}} + x_{\tilde{r}_2} - x_{t^*}\|_{1/2}.$$

For the prediction score of $g_\theta(s^*, r^*, t^*)$, according to Triangle Inequalities, we have,

$$\begin{aligned} g_\theta(s^*, r^*, t^*) =& -\|x_{s^*} + x_{r^*} - x_{t^*}\|_{1/2} \\ =& -\|x_{s^*} + x_{r^*} - x_{t^*} + x_{\tilde{r}_1} + x_{\tilde{r}_2} - x_{\tilde{r}_1} \\ & - x_{\tilde{r}_2} + x_{\tilde{e}} - x_{\tilde{e}}\|_{1/2} \\ =& -\|x_{s^*} + x_{\tilde{r}_1} - x_{\tilde{e}} + x_{\tilde{e}} + x_{\tilde{r}_2} - x_{t^*} \\ & + x_{r^*} - x_{\tilde{r}_1} - x_{\tilde{r}_2}\|_{1/2} \\ \geq& g_\theta(s^*, \tilde{r}_1, \tilde{e}) + g_\theta(\tilde{e}, \tilde{r}_2, t^*) \\ & - \|x_{r^*} - x_{\tilde{r}_1} - x_{\tilde{r}_2}\|_{1/2} \\ =& \delta_1 + \delta_2 - \|x_{r^*} - x_{\tilde{r}_1} - x_{\tilde{r}_2}\|_{1/2}. \end{aligned} \tag{A.1}$$

where $\delta_1 = g_\theta(s^*, \tilde{r}_1, \tilde{e})$, $\delta_2 = g_\theta(\tilde{e}, \tilde{r}_2, t^*)$, and $\|\|_{1/2}$ denotes L_1 or L_2 norm.

Consider all indicative path $j \in [1, n]$, we can get,

$$\begin{aligned} g_\theta(s^*, r^*, t^*) =& -\|x_{s^*} + x_{r^*} - x_{t^*}\|_{1/2} \\ \geq& \max_{1 \leq j \leq n} (-\|x_{r*} - x_{\tilde{r}_1^{(j)}} - x_{\tilde{r}_2^{(j)}}\|_{1/2} + \delta_1^{(j)} + \delta_2^{(j)}). \end{aligned}$$

□

Proof of Theorem 2.

Proof. According to the property of gaussian distribution, we have $g_e - g_\theta \sim \mathcal{N}(\mu_1 - \mu_0, \sigma_0^2 + \sigma_1^2)$. Then the probability $p(g_e - g_\theta > 0)$ can be calculated as,

$$p(g_e - g_\theta > 0) = 1 - \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi(\sigma_0^2 + \sigma_1^2)}} e^{-\frac{(x - \mu_1 + \mu_0)^2}{2(\sigma_0^2 + \sigma_1^2)}}, \tag{A.2}$$

$$= 1 - \Phi(\frac{\mu_0 - \mu_1}{\sqrt{\sigma_0^2 + \sigma_1^2}}). \tag{A.3}$$

Where $\Phi(\cdot)$ is the cumulative distribution function of normal distribution. After this, we have $1 - \Phi(\frac{\mu_0 - \mu_1}{\sqrt{\sigma_0^2 + \sigma_1^2}}) > \delta > \frac{1}{2}$. Then we can get the bound of σ_1 as follows,

$$\Phi(\frac{\mu_0 - \mu_1}{\sqrt{\sigma^2 + \sigma^2}}) < 1 - \delta, \tag{A.4}$$

$$\frac{\mu_0 - \mu_1}{\sqrt{\sigma_1^2 + \sigma_0^2}} < \Phi^{-1}(1 - \delta), \tag{A.5}$$

$$\sigma_1 < \sqrt{\frac{(\mu_0 - \mu_1)^2}{\Phi^{-1}(1 - \delta)^2} - \sigma_0^2}. \tag{A.6}$$

End the proof. □

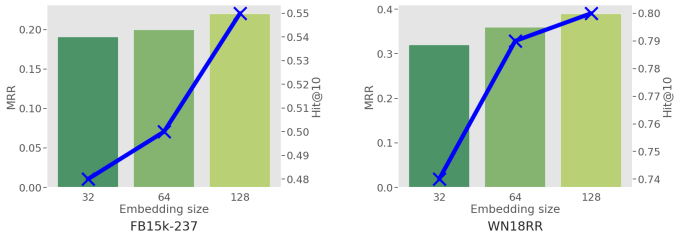

Figure 9: Attack perforance w.r.t. Embedding size of surrogate model. The lines are Hit@10 and the bars are MRR. The results are measured when using TransE to attack ConvE on FB15k-237 and WN18RR.

Table 5: Examples of poisoned triplets generated by proposed attacks.

Target triplet	Poisoned Triplets
(United States of America, second level divisions, Province of Rome)	(United States of America, split to, United States of America)→ (United States of America, locations,Province of Rome)
(Chuck Norris, dated_currency, Wendee Lee)	(Chuck Norris, film_actor_of, Madagascar)→ (Madagascar, film_dated_currency, Wendee Lee)

A.2 More Results

The detailed information of the datasets are shown in Table 4. The generation of the poisoned triplets is depicted in Alg. 1. For more empirical results please refer to Fig. 7, Fig. 9, Fig. 8 and Table. 5.