Positive-Unlabeled Learning with Adversarial Data Augmentation for Knowledge Graph Completion

Zhenwei Tang∗,1, Shichao Pei∗,1, Zhao Zhang2, Yongchun Zhu2, Fuzhen Zhuang3,4, Robert Hoehndorf1, Xiangliang Zhang1,5,1

1King Abdullah University of Science and Technology
2Institute of Computing Technology, Chinese Academy of Sciences
3Institute of Artificial Intelligence, Beihang University
4SKLSDE, School of Computer Science, Beihang University
5University of Notre Dame
{zhenwei.tang, shichao.pei, robert.hoehndorf}@kaust.edu.sa, {zhangzhao2021, zhuyongchun18}@ict.ac.cn, zhuangfuzhen@buaa.edu.cn, xzhang33@nd.edu

Abstract

Most real-world knowledge graphs (KG) are far from complete and comprehensive. This problem has motivated efforts in predicting the most plausible missing facts to complete a given KG, i.e., knowledge graph completion (KGC). However, existing KGC methods suffer from two main issues, 1) the false negative issue, i.e., the sampled negative training instances may include potential true facts; and 2) the data sparsity issue, i.e., true facts account for only a tiny part of all possible facts. To this end, we propose positive-unlabeled learning with adversarial data augmentation (PUDA) for KGC. In particular, PUDA tailors positive-unlabeled risk estimator for the KGC task to deal with the false negative issue. Furthermore, to address the data sparsity issue, PUDA achieves a data augmentation strategy by unifying adversarial training and positive-unlabeled learning under the positive-unlabeled minimax game. Extensive experimental results on real-world benchmark datasets demonstrate the effectiveness and compatibility of our proposed method.

1 Introduction

Knowledge graphs (KG) are of great importance in a variety of real-world applications, such as question answering, recommender systems, and drug discovery [Rossi et al., 2021]. However, it is well-known that even the state-of-the-art KGs still suffer from incompleteness, i.e., many true facts are missing. For example, more than 66% of person entries in Freebase and DBpedia lack their birthplaces [Krompaß et al., 2015]. Such an issue limits the practical applications and has motivated considerable research efforts in the task of knowledge graph completion (KGC) that aims to predict the most plausible missing facts to complete a KG. Formally, a KG is commonly represented as a set of triples in the form of (head entity, relation, tail entity). In this work, we particularly study the problem of predicting the plausibility of each missing head (tail) entity given the relation and the tail (head) entity in a triple, and selecting the most plausible missing head (tail) entities to complete a KG.

Recent years have witnessed increasing interest in the KGC task [Sun et al., 2018; Shikhar et al., 2020; Li et al., 2021]. However, there are common key issues that remain unsolved. (i) The false negative issue. Existing KGC methods require negative samples for model training, and most of them obtain negative samples from nonexistent triples in a KG [Rossi et al., 2021]. However, the nonexistent triples are not necessarily false facts, and some of the nonexistent triples would be added into a KG as true facts with knowledge graph expansion and enrichment. As the example shown in Fig. 1, (Joseph, has job, Professor) and (Joseph, has job, Data Scientist) are true facts in a KG, and it is easy to find that (Joseph, has job, Researcher) is a potential true fact; however, (Joseph, has job, Researcher) is likely to be selected as a negative sample, as it is currently a nonexistent triple. (ii) The data sparsity issue. True facts that can be leveraged as positive samples to guide the training account for only a small portion, e.g., 0.00047% of all possible triples in WN18RR [Dettmers et al., 2018] and 0.00054% in FB15k-237 [Toutanova et al., 2015]. Such extremely sparse positive training data could engender unsatisfactory generalization performance for correctly inferring the missing triples [Pujara et al., 2017].

Figure 1: Nonexistent triples may contain potential true facts.
To address the above issues, we propose a novel method resorting to positive-unlabeled learning with adversarial data augmentation (PUDA) for KGC. **First**, true facts in a KG can be regarded as positive triples, yet the plausibility of nonexistent triples including potential true facts and real false facts is unknown. In this work, we treat each nonexistent triple as an *unlabeled* triple rather than a negative triple used in the most of existing works [Rossi et al., 2021]. Inspired by positive-unlabeled (PU) learning [Bekker and Davis, 2020] that is designed for the learning scenarios where only positive and unlabeled data are given, we propose a novel KGC model resorting to PU learning to circumvent the impact of false negative issue. However, PU learning is originally designed for binary-classification that predicts the exact score of each triple, which is more difficult than necessary for the goal of ranking out the most plausible missing triples desired in KGC task. Therefore, we reformulate the PU risk estimator [Kiryo et al., 2017] and tailor the ordinary PU learning to KGC task. **Second**, as positive triples are extremely rare in KGs, we attempt to augment the positive triples to improve the KGC performance of generalization. Inspired by the recent progress of adversarial training [Mariani et al., 2018] and its capability of generating synthetic data, we develop an adversarial data augmentation strategy to address the data sparsity issue. However, not all the generated synthetic triples are plausible enough to be regarded as positive triples. Therefore, it is more reasonable to treat the synthetic triples as *unlabeled* triples, which could be potential positive or true negative triples. By unifying the adversarial training with the PU risk estimator under PU minimax game, the data sparsity issue would be well mitigated. We summarize the main contributions of this work as: 1) We propose a novel KGC method to circumvent the impact of the false negative issue by resorting to PU learning, and tailoring the ordinary PU risk estimator for KGC task; 2) We design a data augmentation strategy by unifying the idea of adversarial training and PU learning under a PU minimax game to alleviate the data sparsity issue. 3) We conduct extensive experimental evaluations and provide in-depth analysis on real-world benchmark datasets to demonstrate the effectiveness and compatibility of the proposed method.

## 2 Related Work

**Knowledge Graph Completion.** A major line of KGC study focuses on learning distributed representations for entities and relations in KG [Zhang et al., 2020]. (i) Tensor decomposition methods assume the score of a triple can be decomposed into several tensors [Yang et al., 2014; Trouillon et al., 2017; Kazemi and Poole, 2018]; (ii) Geometric methods regard a relation as the translation from a head entity to a tail entity [Bordes et al., 2013; Zhang et al., 2019a; Sun et al., 2018]; (iii) Deep learning methods [Nguyen et al., 2018; Dettmers et al., 2018; Shikhar et al., 2020] utilize deep neural networks to embed KGs. These primary works mainly focus on designing and learning a scoring function to predict the plausibility of a triple. However, our primary focus is beyond the design of the scoring function, aiming at proposing a general KGC framework to address the false negative issue and data sparsity issue.

**Positive-Unlabeled (PU) Learning.** PU learning is a learning paradigm for the setting where a learner can only access to positive and unlabeled data, and the unlabeled data include both positive and negative samples [Bekker and Davis, 2020]. PU learning roughly includes: (1) two-step solutions [Liu et al., 2002; He et al., 2018], (2) methods that consider the unlabeled samples as negative samples with label noise [Shi et al., 2018] (3) unbiased risk estimation based methods [Du Plessis et al., 2014; Christoffel et al., 2016; Xu and Denil, 2019]. Our work is related to the unbiased risk estimator that minimize the expected classification risk to obtain an empirical risk minimizer. However, the original PU risk estimator that based on pointwise ranking loss needs to be specially tailored for our KGC task based on pairwise ranking.

**Adversarial Training.** The adversarial training paradigm [Goodfellow et al., 2014] that trains a generator and a discriminator by playing an adversarial minimax game is originally proposed to generate samples in a continuous space such as images. Several adversarial training methods [Mariani et al., 2018] have been proposed for data augmentation in all classes, rather than specially augmenting positive and unlabeled data. A few works [Wang et al., 2017; Cai and Wang, 2018; Wang et al., 2018] utilize adversarial training for the KGC task. These approaches focus on selecting negative samples that are more useful for model training, while our proposed method generates synthetic samples for data augmentation.

### 3 Preliminaries

**Problem Formulation.** A KG is formulated as 

\[ K = \{(h, r, t) \in E \times R \times E \} \] where \( h, r, t \) denote the head entity, relation, and tail entity in triple \( \langle h, r, t \rangle \), respectively. \( E \) and \( R \) refer to the entity set and the relation set in \( K \). \(|K|\) denotes the total number of triples in \( K \). The KGC problem is to infer the most plausible missing triple from \((\{h, r, t\} | h \in E \land (h, r, t) \notin K \}) \) for each incomplete triple \((h, r, ?)\) (or \((?, r, t)\)), e.g., reporting the top-\(k\) plausible missing triples.

**Scoring Function.** The core of KGC is to learn a scoring function \( \phi(s; \Theta) \) to precisely estimate the plausibility of any triple \( s = \langle h, r, t \rangle \), where \( \Theta \) represents the learnable parameters of an arbitrary scoring function. The main focus of this work is to design a novel framework for KGC to solve the aforementioned two issues rather than developing a novel scoring function. Therefore, we employ the basic yet effective DistMult [Yang et al., 2014] as scoring function. Given the embedding dimension \( d \), head entity embedding \( h \in \mathbb{R}^d \), relation embedding \( r \in \mathbb{R}^{d \times d} \), and tail embedding \( t \in \mathbb{R}^d \), the scoring function is then computed as a bilinear product:

\[
\phi(s; \Theta) = h \times r \times t,
\] (1)

where symbol \( \times \) denotes matrix product, \( r \) is forced to be a diagonal matrix, \( \Theta = \{E_r, E_e\} \) represents learnable parameters, \( E_e \in \mathbb{R}^{|E| \times d} \) and \( E_r \in \mathbb{R}^{R \times d} \), denote the embedding matrices of entities and relations, respectively.
**PU Triple Preparation.** The widely used loss functions [Cai and Wang, 2018] in KGC always require positive triples and negative triples for optimization. The true facts in $K$ can be naturally regarded as **positive** triples. To obtain negative triples, the common practice [Rossi et al., 2021] is to corrupt $h_p$ or $t_p$ in a positive triple $(h_p, r_p, t_p)$. Here we only describe the case of tail corruption; head corruption is defined analogously. Specifically, let $s^i_p = \langle h^i_p, r^i_p, t^i_p \rangle$ be the $i$-th positive triple in $K$. For a given $s^i_p$, we can obtain $N$ corrupted triples $S^i_p = \{\langle h^i_p, r^i_p, t^i_p \rangle\}_N$ by sampling $N$ entities $\{t^i_p \in E \wedge \langle h^i_p, r^i_p, t \in K \}$ and replacing $t^i_p$ by each sampled entity $t^i_p$. Note that the corrupted triples may be potential true facts as illustrated in Fig. 1 and lead to the false negative issue. Therefore, unlike previous work that treats the corrupted triples as negative [Rossi et al., 2021], we regard each triple in $S^i_p$ as an **unlabeled** triple.

**Data Augmentation.** Besides the positive and unlabeled triples, we also generate a set of synthetic triples for data augmentation. Specifically, for a positive triple $s^i_p = \langle h^i_p, r^i_p, t^i_p \rangle$, we generate $M$ synthetic triples $S^i_p = \{\langle h^i_p, r^i_p, t^{i,m}_p \rangle\}_M$ by generating $M$ synthetic entities $\{t^{i,m}_p \in E \}$ via adversarial training and replacing $t^i_p$ by each synthetic entity $t^{i,m}_p$. Similarly, the head entity can be replaced in the same way. Note that not all the generated synthetic triples are plausible enough to be regarded as positive triples. Therefore, we treat each triple in $S^i_p$ as an **unlabeled** triple as well.

## 4 The Proposed Method

As shown in Fig. 2, our proposed method includes two main components. One is the positive-unlabeled (PU) learning for KGC that aims to circumvent the impact of false negative issue. The other is adversarial data augmentation for data sparsity issue. They are unified under a PU minimax game.

### 4.1 Positive-Unlabeled Learning for KGC

Motivated by the aforementioned false negative issue, we aim to design a learning strategy to circumvent the impact of false negative samples. Since KGs only contain true facts (positive triples), the real labels of all nonexistent triples which could be positive or negative are unknown for KG models. Inspired by PU learning that is designed for the learning scenario where only positive data and unlabeled data are given, we denote the nonexistent triples as unlabeled triples, and propose a novel KGC model resorting to PU learning to avoid negative samples. However, prevailing PU learning methods focus on binary-classification, while the final goal of KGC is to provide an optimal ordering of missing triples and selecting the most plausible ones to complete a KG. The binary classification in fact takes a pointwise ranking loss function, which is a more difficult problem than necessary. Pairwise ranking approaches work better in practice than pointwise approaches because predicting relative order is closer to the nature of ranking than predicting class label or relevance score, especially for information retrieval and recommendation tasks [Rendle et al., 2009; Melnikov et al., 2016; Guo et al., 2020]. Therefore, PU learning methods need to be specially tailored for the KGC task with pairwise ranking loss. In this section, we first briefly introduce the PU risk estimator in the classification scenario, then we tailor the PU risk estimator for KGC task.

**PU Classification Risk Estimator.** In positive-negative learning, the empirical risk estimator $R_{pn}(\psi)$ w.r.t. a positive class prior $\pi_p = 1 - \pi_n$ (the percentage of positive samples in all possible samples) is defined as:

$$R_{pn}(\psi) = \pi_p R^+(\psi) + \pi_n R^- (\psi),$$

where $\psi(\cdot)$ is an arbitrary decision function, $R^+(\psi)$ is the expected risk of predicting positive samples as positive (+), and $R^- (\psi)$ is the expected risk of predicting negative samples as negative (-). In positive-unlabeled (PU) learning, due to the absence of negative samples, we cannot directly estimate $R^- (\psi)$. Following [Du Plessis et al., 2015], we define $\pi_n R^- (\psi)$ as:

$$\pi_n R^- (\psi) = R^- (\psi) - \pi_p R^+ (\psi),$$

where $R^- (\psi)$ is the expected risk of predicting unlabeled samples as negative (-) and $R^+ (\psi)$ is the expected risk of predicting positive samples as negative (-). By replacing the second term in Eq.(2) with Eq.(3), the empirical risk estimator can be defined as:

$$R_{pu}(\psi) = \pi_p R^+(\psi) + \pi_u R^-(\psi) - \pi_p R^+ (\psi).$$

Most recent works on risk estimation based PU learning [Kiryo et al., 2017; Akujobi et al., 2020] note that the model tends to suffer from overfitting on the training data when the decision function $\psi$ is complex. Thus, a non-negative risk estimator [Kiryo et al., 2017] is proposed to alleviate the overfitting problem:

$$\hat{R}(\psi) = \pi_p R^+(\psi) + \max\{0, R^- (\psi) - \pi_p R^+ (\psi)\}.$$  

$\hat{R}(\psi)$ should be minimizied to learn the function $\psi$. Note that by introducing the PU risk estimator in Eq. (4) and Eq. (5), we can circumvent the impact of the false negative issue by using unlabeled samples instead of negative samples.

**PU Risk Estimator for KGC.** To define the PU risk estimator for the KGC problem, we first define the scoring function in Eq.(1) as the arbitrary decision function $\psi(\cdot)$ to predict the plausibility of any triple:

$$\psi(\cdot) = \phi(\cdot; \Theta).$$
We replace $\psi(\cdot)$ by $\phi(\cdot)$ in the later descriptions. In the scenario of KGC, the key challenge is to give suitable definitions to the risks $\hat{R}_p^+(\phi)$, $\hat{R}_p^-(\phi)$, and $\hat{R}_u^-(\phi)$. Prevailing PU learning methods [Du Plessis et al., 2014; Kiryo et al., 2017] define the above risks for binary classification as follows:

$$\hat{R}_p^+(\phi) = -\frac{1}{|K|} \sum_{i=1}^{|K|} \ln \sigma (\phi(s_i^p; \Theta)),$$

$$\hat{R}_p^-(\phi) = -\frac{1}{|K|} \sum_{i=1}^{|K|} \ln (\phi(s_i^p; \Theta)),$$

$$\hat{R}_u^-(\phi) = -\frac{1}{|K|} \sum_{i=1}^{|K|} \sum_{j=1}^N \ln (\phi(s_{ij}^u; \Theta)),$$

where $\ln \sigma(\cdot)$ is the log-sigmoid function and is a commonly used function for risk estimator [Kiryo et al., 2017], and $s_{ij}^u \in S_i^u$ denotes an unlabeled triple w.r.t. $s_i^p$. The output values of the score function $\phi(s; \Theta)$ can differentiate the positive triples from the unlabeled triples. However, it is unnecessarily hard to push $\sigma(\phi(s; \Theta))$ to be 1 (0) for all positive (unlabeled) triples in a KGC task. It is easier and more desirable to just push positive triples to be ranked higher than the unlabeled triples.

Let $s_i^p > s_{ij}^u$ be the optimal ordering of a given positive triple $s_i^p$ and unlabeled triples $S_i^u$, i.e., $s_i^p$ is ranked higher than every triple in $S_i^u$. Without loss of generality, we can assume that all unlabeled triples in $S_i^u$ are independent given that they have the same head and relation (or the same tail and relation) from the positive triple. The optimization of $\Theta$ towards the optimal ordering $s_i^p$ can then be formulated on each pair of triples $(s_i^p, s_{ij}^u)$ [Rendle et al., 2009] as follows:

$$\ln p(s_i^p|\Theta) = \ln \prod_{j=1}^N \sigma (\phi(s_i^p; \Theta) - \phi(s_{ij}^u; \Theta))$$

$$= \sum_{j=1}^N \ln \sigma (\phi(s_i^p; \Theta) - \phi(s_{ij}^u; \Theta)),$$

where maximizing the difference between $\phi(s_i^p; \Theta)$ and $\phi(s_{ij}^u; \Theta)$ would push the positive triple $s_i^p$ to rank higher than all unlabeled triples in $S_i^u$.

The lack of pairwise triples in Eq.(9) hinders the introduction of pairwise comparison. It focuses only on minimizing the risk of predicting $\phi(s_{ij}^u; \Theta)$ as a too large value. To incorporate the pairwise orderings, we can replace $\ln \sigma(\phi(s_{ij}^u; \Theta))$ in Eq.(9) by $\ln \sigma (\phi(s_i^p; \Theta) - \phi(s_{ij}^u; \Theta))$. That is, measuring the likelihood of $s_{ij}^u$ being a small value is replaced by measuring the likelihood of $s_{ij}^u$ being just smaller than its positive counterpart $s_i^p$. Then, Eq.(9) becomes

$$\hat{R}_u^-(\phi) = \frac{1}{|K|} \sum_{i=1}^{|K|} \frac{1}{N} \sum_{j=1}^N \ln \sigma (\phi(s_i^p; \Theta) - \phi(s_{ij}^u; \Theta)),$$

where $\hat{R}_u^-(\phi)$ is the risk of obtaining a non-optimal ordering of the positive triple $s_i^p$ and the unlabeled triples $S_i^u$. It is worth noting that Eq.(11) makes no penalty on the ranking relations among unlabeled triples, unlike Eq.(9) which targets to score all unlabeled triples to be the same low values. Hence, Eq.(11) allows some unlabeled triples to be ranked higher than other unlabeled triples, as long as all unlabeled triples are ranked lower than the positive triple. Those high-ranked unlabeled triples have the room to be promoted as positive triples. Combining Eq.(5) with Eq.(7), Eq.(8), and Eq.(11), we have the well defined $\hat{R}(\phi)$ for KGC problem, which embeds pairwise ranking objective in the PU risk estimator to deal with the false negative issue.

**Further Discussion.** The superiority of our proposed PU learning for KGC is that it can handle the false negative issue, which existing pairwise ranking loss based methods still suffer from, because they simply treat all unlabeled data as negative [Rossi et al., 2021]. In other words, pairwise ranking loss aims to rank positive triples ahead unlabeled triples. However, unlabeled data contain both positive and negative samples. Positive triples thus should not be ranked ahead the false negative (positive) triples within unlabeled data. Recent research shows that the bias can be canceled by particular PU optimization objectives [Kiryo et al., 2017]. Specifically, Eq.(5) is the PU objective and Eq.(8) serves as the term to cancel the bias. They reflect the motivation of using PU learning to alleviate the false negative problem.

### 4.2 Adversarial Data Augmentation

Inspired by the recent progress of adversarial training [Mariani et al., 2018], we aim to use adversarial data augmentation to alleviate the long-existing data sparsity issue. Since positive triples only account for a tiny portion of all possible triples, we follow the principle of adversarial training to generate synthetic triples $S_i^*$ for data augmentation rather than only using authentic triples $\{(h, r, t)|h \in E \land r \in R \land t \in \hat{E}\}$. However, not all the generated synthetic triples are plausible enough to be regarded as positive triples. A more reasonable way is to treat them as unlabeled triples. With the well-defined PU risk estimator for KGC, we can train generator $G$ and discriminator $D$ by playing a PU minimax game, in which $G$ tries to generate plausible synthetic triples to fool $D$ and $D$ tries to distinguish if a triple is synthetic or authentic using the proposed PU risk estimator. We explain the detail of adversarial data augmentation as follows.

**Minimax Game.** For one positive triple $s_i^p = (h_i^p, r_i^p, t_i^p)$, a synthetic triple $s_{im}^* = (h_{im}^p, r_{im}^p, t_{im}^p)$ can be generated using a multi-layer perceptron (MLP) given a random noise $z_{im}$:

$$G(z; \Omega) = \text{Tanh}(W_2 \cdot \text{ReLU}(W_1 \cdot z + b_1) + b_2),$$

$$z_{im} \sim \mathcal{N}(0, \delta I),$$

$$t_{im}^* = G(z_{im}; \Omega),$$

$$s_{im}^* = (h_{im}^p, r_{im}^p, t_{im}^*),$$

where $\delta$ is the mean of noise input with the same size as the embedding dimension $d$, and $I \in \mathbb{R}^{d \times d}$ is the identity matrix whose magnitude is controlled by the deviation of the noise input $\delta$. $\Omega = \{W_1, W_2, b_1, b_2\}$ includes all learnable weights and biases of the two-layer MLP.

Given a generated synthetic triple $s_{im}^* \in S_i^*$, the discriminator $D$ tries to predict the plausibility of $s_{im}^*$ to distinguish if
it is authentic or synthetic. Note that scoring function $\phi(s; \Theta)$ also predicts the plausibility of each triple. Thus, the discriminator can be defined as calculating the scoring function:

$$D(s; \Theta) = \phi(s; \Theta).$$  \hfill (13)

Since positive triples only account for a tiny portion of all authentic triples, the goal is to generate synthetic triples to augment positive triples. Therefore, the adversarial objective to train $G$ and $D$ can be designed as the minimax game w.r.t. positive authentic triples and synthetic triples:

$$\min_D \max_G : \frac{1}{|K|} \sum_{i=1}^{|K|} \left( - \frac{1}{M} \sum_{m=1}^M \ln \sigma(\phi(s^p_i; \Theta) - \phi(s^s_i; \Theta)) \right).$$  \hfill (14)

With a fixed $G$, by minimizing Eq.(14), $D$ is trained to optimize the relative ordering of the generated synthetic triple $s^s_i$ and a generated synthetic triple $s^p_i$, such that $s^s_i$ ranks higher than $s^p_i$. On the contrary, with a fixed $D$, $G$ is trained by maximizing Eq. (14) to fool $D$, such that $s^s_i$ ranks higher than or close to $s^p_i$.

**PU Minimax Game.** Note that with the minimax game designed as Eq.(14), the generated synthetic triples are treated as authentic triples. Therefore, the adversarial objective to train $G$ and $D$ can be designed as the minimax game w.r.t. positive authentic triples and synthetic triples:

$$\min_D \max_G : \frac{1}{|K|} \sum_{i=1}^{|K|} \left( - \frac{1}{M} \sum_{m=1}^M \ln \sigma(\phi(s^p_i; \Theta) - \phi(s^s_i; \Theta)) \right).$$  \hfill (14)

With a fixed $G$, by minimizing Eq.(14), $D$ is trained to optimize the relative ordering of the given positive triple $s^p_i$ and a generated synthetic triple $s^s_i$, such that $s^s_i$ ranks higher than $s^p_i$. On the contrary, with a fixed $D$, $G$ is trained by maximizing Eq. (14) to fool $D$, such that $s^s_i$ ranks higher than or close to $s^p_i$.

**PU Minimax Game.** Note that with the minimax game designed as Eq.(14), the generated synthetic triples are treated as authentic triples. Therefore, the adversarial objective to train $G$ and $D$ can be designed as the minimax game w.r.t. positive authentic triples and synthetic triples:

$$\min_D \max_G : \frac{1}{|K|} \sum_{i=1}^{|K|} \left( - \frac{1}{M} \sum_{m=1}^M \ln \sigma(\phi(s^p_i; \Theta) - \phi(s^s_i; \Theta)) \right).$$  \hfill (14)

With a fixed $G$, by minimizing Eq.(14), $D$ is trained to optimize the relative ordering of the generated synthetic triple $s^s_i$ and a generated synthetic triple $s^p_i$, such that $s^s_i$ ranks higher than $s^p_i$. On the contrary, with a fixed $D$, $G$ is trained by maximizing Eq. (14) to fool $D$, such that $s^s_i$ ranks higher than or close to $s^p_i$.

**PU Minimax Game.** Note that with the minimax game designed as Eq.(14), the generated synthetic triples are treated as authentic triples. Therefore, the adversarial objective to train $G$ and $D$ can be designed as the minimax game w.r.t. positive authentic triples and synthetic triples:

$$\min_D \max_G : \frac{1}{|K|} \sum_{i=1}^{|K|} \left( - \frac{1}{M} \sum_{m=1}^M \ln \sigma(\phi(s^p_i; \Theta) - \phi(s^s_i; \Theta)) \right).$$  \hfill (14)

With a fixed $G$, by minimizing Eq.(14), $D$ is trained to optimize the relative ordering of the given positive triple $s^p_i$ and a generated synthetic triple $s^s_i$, such that $s^s_i$ ranks higher than $s^p_i$. On the contrary, with a fixed $D$, $G$ is trained by maximizing Eq. (14) to fool $D$, such that $s^s_i$ ranks higher than or close to $s^p_i$.

**PU Minimax Game.** Note that with the minimax game designed as Eq.(14), the generated synthetic triples are treated as authentic triples. Therefore, the adversarial objective to train $G$ and $D$ can be designed as the minimax game w.r.t. positive authentic triples and synthetic triples:

$$\min_D \max_G : \frac{1}{|K|} \sum_{i=1}^{|K|} \left( - \frac{1}{M} \sum_{m=1}^M \ln \sigma(\phi(s^p_i; \Theta) - \phi(s^s_i; \Theta)) \right).$$  \hfill (14)

With a fixed $G$, by minimizing Eq.(14), $D$ is trained to optimize the relative ordering of the given positive triple $s^p_i$ and a generated synthetic triple $s^s_i$, such that $s^s_i$ ranks higher than $s^p_i$. On the contrary, with a fixed $D$, $G$ is trained by maximizing Eq. (14) to fool $D$, such that $s^s_i$ ranks higher than or close to $s^p_i$.

**PU Minimax Game.** Note that with the minimax game designed as Eq.(14), the generated synthetic triples are treated as authentic triples. Therefore, the adversarial objective to train $G$ and $D$ can be designed as the minimax game w.r.t. positive authentic triples and synthetic triples:

$$\min_D \max_G : \frac{1}{|K|} \sum_{i=1}^{|K|} \left( - \frac{1}{M} \sum_{m=1}^M \ln \sigma(\phi(s^p_i; \Theta) - \phi(s^s_i; \Theta)) \right).$$  \hfill (14)

With a fixed $G$, by minimizing Eq.(14), $D$ is trained to optimize the relative ordering of the generated synthetic triple $s^s_i$ and a generated synthetic triple $s^p_i$, such that $s^s_i$ ranks higher than $s^p_i$. On the contrary, with a fixed $D$, $G$ is trained by maximizing Eq. (14) to fool $D$, such that $s^s_i$ ranks higher than or close to $s^p_i$.

**PU Minimax Game.** Note that with the minimax game designed as Eq.(14), the generated synthetic triples are treated as authentic triples. Therefore, the adversarial objective to train $G$ and $D$ can be designed as the minimax game w.r.t. positive authentic triples and synthetic triples:

$$\min_D \max_G : \frac{1}{|K|} \sum_{i=1}^{|K|} \left( - \frac{1}{M} \sum_{m=1}^M \ln \sigma(\phi(s^p_i; \Theta) - \phi(s^s_i; \Theta)) \right).$$  \hfill (14)

With a fixed $G$, by minimizing Eq.(14), $D$ is trained to optimize the relative ordering of the given positive triple $s^p_i$ and a generated synthetic triple $s^s_i$, such that $s^s_i$ ranks higher than $s^p_i$. On the contrary, with a fixed $D$, $G$ is trained by maximizing Eq. (14) to fool $D$, such that $s^s_i$ ranks higher than or close to $s^p_i$.

**PU Minimax Game.** Note that with the minimax game designed as Eq.(14), the generated synthetic triples are treated as authentic triples. Therefore, the adversarial objective to train $G$ and $D$ can be designed as the minimax game w.r.t. positive authentic triples and synthetic triples:

$$\min_D \max_G : \frac{1}{|K|} \sum_{i=1}^{|K|} \left( - \frac{1}{M} \sum_{m=1}^M \ln \sigma(\phi(s^p_i; \Theta) - \phi(s^s_i; \Theta)) \right).$$  \hfill (14)

With a fixed $G$, by minimizing Eq.(14), $D$ is trained to optimize the relative ordering of the given positive triple $s^p_i$ and a generated synthetic triple $s^s_i$, such that $s^s_i$ ranks higher than $s^p_i$. On the contrary, with a fixed $D$, $G$ is trained by maximizing Eq. (14) to fool $D$, such that $s^s_i$ ranks higher than or close to $s^p_i$.
Table 1: Experimental results on FB15k-237 and WN18RR. The best results are in boldface, the strongest baseline performance is underlined.

| Category          | Model          | FB15k-237 |       |       |       |       | WN18RR |       |       |       |
|-------------------|----------------|-----------|-------|-------|-------|-------|--------|-------|-------|-------|
|                   | Model          | MRR       | Hit@1 | Hit@3 | Hit@10| MRR   | Hit@1 | Hit@3 | Hit@10|
| Tensor Decomposition | DistMult     | 0.313     | 0.224 | 0.275 | 0.490 | 0.433 | 0.397 | -     | -     | 0.502 |
|                   | ComplEx*      | 0.247     | 0.158 | 0.344 | 0.428 | 0.440 | 0.410 | 0.460 | 0.510 |
|                   | SimpleIE*     | 0.179     | 0.100 | -     | -     | 0.344 | 0.398 | 0.383 | -     | 0.427 |
| Geometric         | TransE        | 0.310     | 0.217 | -     | 0.497 | 0.206 | 0.279 | -     | -     | 0.495 |
|                   | CrossE*       | 0.298     | 0.212 | -     | 0.471 | 0.405 | 0.381 | -     | -     | 0.450 |
|                   | RotatE*       | 0.338     | 0.241 | 0.375 | 0.533 | 0.476 | 0.428 | 0.492 | 0.571 |
| Deep Learning     | CompE*        | 0.244     | 0.337 | 0.356 | 0.501 | 0.430 | 0.490 | 0.440 | 0.520 |
|                   | ConvKB*       | 0.230     | 0.140 | -     | 0.415 | 0.249 | 0.056 | -     | 0.525 |
|                   | CompGNC*      | 0.355     | 0.264 | 0.390 | 0.535 | 0.479 | 0.443 | 0.494 | 0.546 |
|                   | HRAN          | 0.355     | 0.263 | 0.390 | 0.541 | 0.479 | 0.450 | 0.494 | 0.542 |
| Negative Sampling | KBGan*        | 0.278     | -     | -     | 0.458 | 0.214 | -     | -     | 0.472 |
|                   | NSCaching     | 0.302     | -     | -     | 0.481 | 0.443 | -     | -     | 0.518 |
|                   | SANS          | 0.336     | -     | -     | 0.531 | 0.476 | -     | -     | 0.573 |
|                   | PUDA          | 0.369     | 0.268 | 0.408 | 0.578 | 0.481 | 0.436 | 0.498 | 0.582 |

Table 2: Ablation study on FB15k-237 dataset.

|        | MRR   | Hit@1 | Hit@3 | Hit@10|
|--------|-------|-------|-------|-------|
| PN     | 0.313 | 0.224 | -     | 0.490 |
| PU-C   | 0.303 | 0.220 | 0.333 | 0.481 |
| PU-R   | 0.360 | 0.260 | 0.398 | 0.566 |
| DA     | 0.342 | 0.243 | 0.380 | 0.547 |
| PUDA   | 0.369 | 0.268 | 0.408 | 0.578 |

Table 3: PUDA upon different scoring functions on FB15k-237.

|        | MRR   | Hit@1 | Hit@3 | Hit@10|
|--------|-------|-------|-------|-------|
| DistMult | 0.313 | 0.224 | -     | 0.490 |
| PUDA    | 0.369 | 0.268 | 0.408 | 0.578 |
| TransE  | 0.310 | 0.217 | -     | 0.497 |
| PUDA-TransE | 0.340 | 0.227 | 0.391 | 0.560 |
| ConvKB  | 0.230 | 0.140 | -     | 0.415 |
| PUDA-ConvKB | 0.315 | 0.213 | 0.351 | 0.520 |

5.3 Ablation Study (RQ2)

We conduct ablation study to show the effectiveness of each component of PUDA to answer RQ2. In Table 2, PN denotes the original DistMult model trained with ordinary positive-negative training objective by regarding all unlabeled triples as negative data. PU-C and PU-R denote the ordinary binary-classification-based PU learning (pointwise ranking) and our proposed pairwise ranking-based PU learning, respectively. DA denotes the DistMult model only incorporating the proposed adversarial data augmentation. We have the following observations from Table 2. (i) PU-R shows the consistently superior performance than PN, demonstrating the effectiveness of our proposed PU learning for KGC by regarding negative triples as unlabeled triples to circumvent the impact of false negative issue. (ii) DA achieves a substantial improvement over PN, showing that the generated synthetic triples by our proposed adversarial data augmentation play a crucial role in improving the performance of KGC. (iii) PU-R outperforms PU-C, presenting the advantage of the proposed pairwise ranking-based PU risk estimator over the ordinary binary-classification-based PU risk estimator. We believe that it is because the pointwise ranking is unnecessarily hard for KGC task and harms the performance of KGC models. In addition, PU-C performs slightly worse than PN, showing that the ordinary binary-classification-based PU learning harms the performance of KGC model and is not suitable for the KGC task, also revealing the necessity of our proposed pairwise ranking-based PU risk estimator. (iv) PUDA outperforms not only PN, but also PU-C, PU-R, and DA, demonstrating the effectiveness of unifying adversarial training objective with the PU risk estimator with the PU minimax game.

5.4 Compatibility (RQ3)

We implement PUDA upon well-established scoring functions. As Table 3 shown, our proposed method can achieve a remarkable improvement over the original scoring functions on all metrics. Note that the included scoring functions are representative methods of tensor decomposition, geometric, and deep learning based methods. Such results demonstrate that PUDA is compatible with a wide range of scoring functions and could be a plug-and-play component for existing methods and we leave the potential of PUDA upon more advanced scoring functions to be discovered.

6 Conclusion

In this paper, we proposed the novel PUDA for KGC, which utilizes PU risk estimator and adversarial data augmentation to deal with the false negative issue and data sparsity issue, respectively. In particular, we tailored the ordinary PU risk estimator for KGC task and seamlessly unified the PU risk estimator with adversarial training objective via the PU minimax game. Extensive experimental results demonstrate the effectiveness and compatibility of PUDA.

Acknowledgments

This work has been supported by funding from King Abdullah University of Science and Technology under Award No. BAS/1/1635-01-01, URF/1/4355-01-01, URF/1/4675-01-01, and FCC/1/1976-34-01. This work is also supported by the National Key Research and Development Program of China under Grant No.2021ZD0113602, the National Natural Science Foundation of China under Grant No.62176014.
References

[Ahrabian et al., 2020] K. Ahrabian, A. Feizi, Y. Salehi, W. Hamilton, and Avishek J. Bose. Structure aware negative sampling in knowledge graphs. In EMNLP, pages 6093–6101, 2020.

[Akjujuobi et al., 2020] U. Akjujuobi, J. Chen, M. Elhoseiny, M. Spranger, and X. Zhang. Temporal positive-unlabeled learning for biomedical hypothesis generation via risk estimation. NeurIPS, 33:4597–4609, 2020.

[Bekker and Davis, 2020] J. Bekker and J. Davis. Learning from positive and unlabeled data: A survey. Machine Learning, 109(4):719–760, 2020.

[Bordes et al., 2013] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. NeurIPS, 26, 2013.

[Breit et al., 2020] A. Breit, S. Ott, A. Agibetov, and M. Samwald. Openbiolink: a benchmarking framework for large-scale biomedical link prediction. Bioinformatics, 36(13):4097–4098, 2020.

[Cai and Wang, 2018] L. Cai and William Y. Wang. Kbgan: Adversarial learning for knowledge graph embeddings. In NAACL, pages 1470–1480, 2018.

[Christoffel et al., 2016] M. Christoffel, G. Niu, and M. Sugiyama. Class-prior estimation for learning from positive and unlabeled data. In ACMEL, pages 221–236. PMLR, 2016.

[Dettmers et al., 2018] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. Convolutional 2d knowledge graph embeddings. In AAAI, 2018.

[Du Plessis et al., 2014] M. Du Plessis, G. Niu, and M. Sugiyama. Analysis of learning from positive and unlabeled data. NeurIPS, 27:703–711, 2014.

[Du Plessis et al., 2015] M. Du Plessis, G. Niu, and M. Sugiyama. Convex formulation for learning from positive and unlabeled data. In ICML, pages 1386–1394. PMLR, 2015.

[Goodfellow et al., 2014] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. NeurIPS, 27, 2014.

[Guo et al., 2020] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zaman, C. Wu, W. Croft, and X. Cheng. A deep look into neural ranking models for information retrieval. Information Processing & Management, 57(6):102067, 2020.

[He et al., 2018] F. He, T. Liu, G. Webb, and D. Tao. Instance-dependent pu learning by bayesian optimal relabeling. arXiv preprint arXiv:1808.02180, 2018.

[Kazemi and Poole, 2018] S. Kazemi and D. Poole. Simple embedding for link prediction in knowledge graphs. In NeurIPS, pages 4289–4300, 2018.

[Kiryo et al., 2017] R. Kiryo, G. Niu, M. du Plessis, and M. Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In NeurIPS, pages 1674–1684, 2017.

[Krompaß et al., 2015] D. Krompaß, S. Baier, and V. Tresp. Type-constrained representation learning in knowledge graphs. In ISWC, pages 640–655, 2015.

[Li et al., 2021] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. Xiong. Learning knowledge graph embedding with heterogeneous relation attention networks. TNNLS, 2021.

[Li et al., 2002] B. Liu, W. Lee, P. Yu, and X. Li. Partially supervised classification of text documents. In ICML, volume 2, pages 387–394. Sydney, NSW, 2002.

[Mariani et al., 2018] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi. Bagan: Data augmentation with balancing gan. arXiv preprint arXiv:1803.09655, 2018.

[Melnikov et al., 2016] V. Melnikov, P. Gupta, B. Frick, D. Kaiman, and E. Hüllemeier. Pairwise versus pointwise ranking: A case study. Schedae Informaticae, 2016.

[Nguyen et al., 2018] T. Nguyen, D. Nguyen, D. Phung, et al. A novel embedding model for knowledge base completion based on convolutional neural network. In IAAI, pages 327–333, 2018.

[Pujara et al., 2017] J. Pujara, E. Augustine, and L. Getoor. Sparsity and noise: Where knowledge graph embeddings fall short. In EMNLP, pages 1751–1756, 2017.

[Rendle et al., 2009] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In UAI, pages 452–461, 2009.

[Rossi et al., 2021] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. TKDD, 15(2):1–49, 2021.

[Shi et al., 2018] H. Shi, S. Pan, J. Yang, and C. Gong. Positive and unlabeled learning via loss decomposition and centroid estimation. In ICAI, pages 2689–2695, 2018.

[Shikhar et al., 2020] V. Shikhar, S. Soumya, N. Vikram, and T. Partha. Composition-based multi-relational graph convolutional networks. In ICLR, 2020.

[Sun et al., 2018] Z. Sun, Z. Deng, J. Nie, and J. Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In ICLR, 2018.

[Toutanova et al., 2015] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon. Representing text for joint embedding of text and knowledge bases. In EMNLP, pages 1499–1509, 2015.

[Trouillon et al., 2017] T. Trouillon, C. Dance, É. Gaussier, J. Welbl, S. Riedel, and G. Bouchard. Knowledge graph completion via complex tensor factorization. JMLR, 18:1–38, 2017.

[Wang et al., 2017] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In SIGIR, pages 515–524, 2017.

[Wang et al., 2018] P. Wang, S. Li, and R. Pan. Incorporating gan for negative sampling in knowledge representation learning. In AAAI, volume 32, 2018.

[Xu and Denil, 2019] D. Xu and M. Denil. Positive-unlabeled reward learning. arXiv preprint arXiv:1911.00459, 2019.

[Yang et al., 2014] B. Yang, W. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575, 2014.

[Zhang et al., 2019a] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen. Interaction embeddings for prediction and explanation in knowledge graphs. In WSDM, pages 96–104, 2019.

[Zhang et al., 2019b] Y. Zhang, Q. Yao, Y. Shao, and L. Chen. Nscaching: simple and efficient negative sampling for knowledge graph embedding. In ICDE, pages 614–625. IEEE, 2019.

[Zhang et al., 2020] Z. Zhang, F. Zhuang, H. Zhu, Z. Shi, H. Xiong, and Q. He. Relational graph neural network with hierarchical attention for knowledge graph completion. In AAAI, volume 34, pages 9612–9619, 2020.