Understanding and Improving Knowledge Graph Embedding for Entity Alignment

Lingbing Guo [*1 2 3] **Qiang Zhang** [*1 2 3] **Zequn Sun** [4] **Mingyang Chen** [1 3] **Wei Hu** [4] **Huajun Chen** [1 2 3]

Abstract

Embedding-based entity alignment (EEA) has recently received great attention. Despite significant performance improvement, few efforts have been paid to facilitate understanding of EEA methods. Most existing studies rest on the assumption that a small number of pre-aligned entities can serve as anchors connecting the embedding spaces of two KGs. Nevertheless, no one has investigated the rationality of such an assumption. To fill the research gap, we define a typical paradigm abstracted from existing EEA methods and analyze how the embedding discrepancy between two potentially aligned entities is implicitly bounded by a predefined margin in the score function. Further, we find that such a bound cannot guarantee to be tight enough for alignment learning. We mitigate this problem by proposing a new approach, named NeoEA, to explicitly learn KG-invariant and principled entity embeddings. In this sense, an EEA model not only pursues the closeness of aligned entities based on geometric distance, but also aligns the *neural ontologies* of two KGs by eliminating the discrepancy in embedding distribution and underlying ontology knowledge. Our experiments demonstrate consistent and significant performance improvement against the best-performing EEA methods.

1. Introduction

Knowledge graphs (KGs), such as DBpedia (Auer et al., 2007) and Wikidata (Vrandečić & Krötzsch, 2014), have become crucial resources for many AI applications. Although a large-scale KG offers structured knowledge derived from millions of facts in the real world, it is still incomplete by nature, and the downstream applications always demand more knowledge. Entity alignment (EA) is then proposed to solve this issue, which exploits the potentially aligned entities among different KGs to facilitate knowledge fusion and exchange (Suchanek et al., 2012).

Recently, embedding-based entity alignment (EEA) methods (Chen et al., 2017; Sun et al., 2017; Zhu et al., 2017; Wang et al., 2018; Guo et al., 2019; Ye et al., 2019; Wu et al., 2019; Sun et al., 2020b; Fey et al., 2020) have been prevailing in the EA area. The central idea is to encode entity/relation semantics into embeddings and estimate entity similarity based on their embedding distance. These methods either learn an alignment function f_a to minimize the distance between the embeddings of a pair of aligned entities (Wang et al., 2018), or directly map these two entities to one vector representation (Sun et al., 2017). Meanwhile, they leverage a shared score function f_s to encode semantics into representations, such that two potentially aligned entities shall have similar feature expression. During this process, a small number of aligned entity pairs (a.k.a., seed alignment) are required as supervision data to align (or merge) the embedding spaces of two KGs.

Different from the conventional methods (Suchanek et al., 2012; El-Roby & Aboulnaga, 2015; Lacoste-Julien et al., 2013) that manually collect and select discriminative features, EEA ones rarely rely on third-party tools or pipeline for preprocessing. Commonly, they just need triples or adjacency matrices of KGs as the input data, and have achieved comparable or better performance on many benchmarks (Sun et al., 2017; 2018; 2020c).

Another strength is that EEA is free of symbolic heterogeneity of two KGs (Sun et al., 2020c). The relationships among entities are interpreted by the score function f_s and manifested as distances in the embedding space. Hence, the similarity between a pair of entities can be defined and estimated smoothly. Without loss of generality, we consider the score function of TransE (Bordes et al., 2013) as f_s. It describes a triple (e_i, r, e_j) by $\mathbf{e}_i + \mathbf{r} \approx \mathbf{e}_j$, where e_i, r, e_j are the head entity, relation, and tail entity, respectively. The

[*]Equal contribution [1]College of Computer Science and Technology, Zhejiang University [2]ZJU-Hangzhou Global Scientific and Technological Innovation Center [3]Alibaba-Zhejiang University Joint Reseach Institute of Frontier Technologies [4]State Key Laboratory for Novel Software Technology, Nanjing University, China. Correspondence to: Qiang Zhang <qiang.zhang.cs@zju.edu.cn>, Huajun Chen <huajunsir@zju.edu.cn >.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

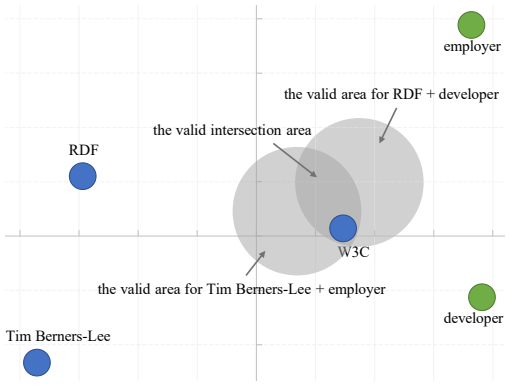


Figure 1: Illustration of margin-based score function. Blue and green nodes denote entity embeddings and relation embeddings, respectively. The center of each grey circle is assigned according to the head entity and relation in the triple. The radii are exactly the margin λ.

approximation is achieved by defining a margin λ to ensure:

$$||\mathbf{e}_i + \mathbf{r} - \mathbf{e}_j|| \leq \lambda, \tag{1}$$

where $||\cdot||$ denotes the L1 or L2 distance. We illustrate this concept in Figure 1, where the two triples *(Tim Berners-Lee, employer, W3C)* and *(RDF, developer, W3C)* have the same tail entity *W3C*. The valid area for *W3C* is decided by two circles. Their centers are *Tim Berners-Lee* + *employer* and *RDF* + *developer*, respectively. The radii are exactly the margin λ. Therefore, the desired embedding of *W3C* should be located in the intersection area.

Many existing studies (Chen et al., 2017; Sun et al., 2017; 2018) have explored how to choose a proper λ for entity alignment, but we argue this goes beyond a mere parameter-tuning problem. In this paper, we define a paradigm leveraged by the current methods. We show that the embedding discrepancy of an underlying aligned entity pair is bounded by $\epsilon \propto \lambda$, for most EEA methods (Chen et al., 2017; Sun et al., 2017; Zhu et al., 2017; Sun et al., 2018; Pei et al., 2019a), or allowing more divergence between two potentially aligned entities (Wang et al., 2018; Guo et al., 2019; Wu et al., 2019; Ye et al., 2019; Sun et al., 2020b). Further, we find that this margin-based bound cannot be as tight as expected, causing minimal constraints on the entities with few neighbors. Take *W3C* in Figure 1 as an example. The valid area will shrink and finally disappear if this entity has more and more linked neighbors. There is only one way to mitigate this problem – enlarging the radii, which will allow more divergence for entities with a few neighbors.

We consider additional constraints on entity embeddings to mitigate the above problem, which we name neural-ontology-driven entity alignment (abbr., NeoEA). In Semantic Web, an *ontology* (Horrocks et al., 2006; Grau et al., 2008; Baader et al., 2005) is usually comprised of *axioms* that define the concepts and relationships for entities and relations. Those axioms make a KG *principled* (i.e., constrained by rules). For example, an "Object Property Domain" axiom in OWL 2 EL (Grau et al., 2008) claims the valid head entities for a specific relation (e.g., the head entities of relation "birthPlace" should be in class "Person"), and it thus determines the head entity distributions of this relation. The *neural ontology* in this paper is reversely deduced from the entity embedding distributions, which is clearly different from the existing methods like OWL2Vec* (Chen et al., 2021) that leverages external ontology data to improve KG embeddings. We expect to align the high-level neural ontologies to diminish the discrepancy of entity embedding distributions and ontology knowledge between two KGs. We conducted experiments to verify the effectiveness of NeoEA with several state-of-the-art baselines. We show that NeoEA can consistently and significantly improve their performance across multiple datasets.

2. Background

In this section, we first introduce the preliminaries and the related works, and then discuss the inherent limitations of current EEA methods.

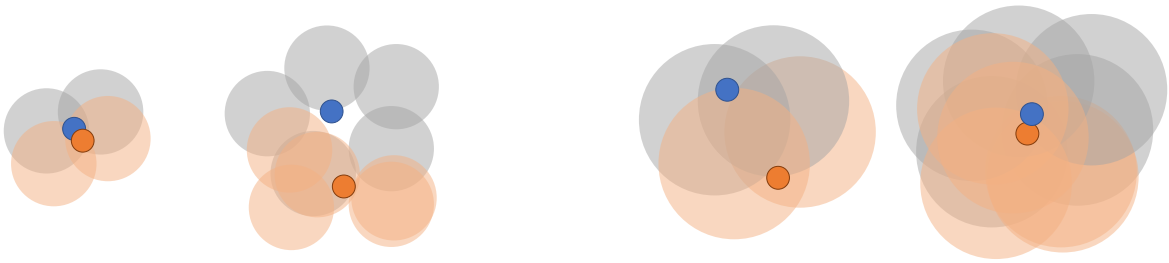
2.1. Embedding-based Entity Alignment

We start by defining a typical KG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ and $\mathcal{R}$ are the entity and relation sets respectively, and $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is the triple set. A triple (e_1, r, e_2) comprises three elements, i.e., the head entity e_1, the relation r, and the tail entity e_2. We use the boldface $\mathbf{e}_1$ to denote the embedding of the entity e_1.

The common paradigm employed by most existing EEA methods (Chen et al., 2017; Sun et al., 2017; Zhu et al., 2017; Sun et al., 2018; Pei et al., 2019a) is then defined as:

Definition 2.1 (Embedding-based Entity Alignment)**.** The input of EEA is two KGs $\mathcal{G}_1 = (\mathcal{E}_1, \mathcal{R}_1, \mathcal{T}_1)$, $\mathcal{G}_2 = (\mathcal{E}_2, \mathcal{R}_2, \mathcal{T}_2)$, and a small subset of aligned entity pairs $\mathcal{S} \subset \mathcal{E}_1 \times \mathcal{E}_2$ as seeds to connect $\mathcal{G}_1$ with $\mathcal{G}_2$. An EEA model consists of two neural functions: an alignment function f_a, which is used to regularize the embeddings of pairwise entities in $\mathcal{S}$; and a score function f_s, which scores the embeddings based on the joint triple set $\mathcal{T}_1 \cup \mathcal{T}_2$. EEA estimates the alignment of an arbitrary entity pair (e_i^1, e_j^2) by their geometric distance $d(\mathbf{e}_i^1, \mathbf{e}_j^2)$.

The existing studies have explored a diversity of f_a. The pioneering work MTransE (Chen et al., 2017) was proposed to learn a mapping matrix to cast an entity embedding $\mathbf{e}_i^1$ to the vector space of $\mathcal{G}_2$. SEA (Pei et al., 2019a) and OTEA (Pei et al., 2019b) extended this approach with adversarial training to learn the projection matrix. Especially, OTEA is highly related to our approach as it is also based on optimal



(a) small margin (b) large margin

Figure 2: Influence of margin λ to different entities in EEA. Blue and orange nodes denote the entities in $\mathcal{E}_1$ and $\mathcal{E}_2$. Grey and orange circles denote the valid area w.r.t. the relevant triples of the two entities, respectively.

transport (OT) (Arjovsky et al., 2017). The differences are: (1) NeoEA provides a general way to align entity embedding distributions, while OTEA regularizes the projection matrix via optimal transport. (2) NeoEA exploits the underlying ontology information to facilitate entity alignment.

Recently, a simpler yet more efficient method was widely-used, which directly maps a pair of aligned entities $(e_i^1, e_i^2) \in \mathcal{S}$ to one embedding vector $\mathbf{e}_i$ (Sun et al., 2017; Zhu et al., 2017; Guo et al., 2019). Meanwhile, some methods (Wang et al., 2018; Pei et al., 2019a; Wu et al., 2019) started to leverage a softer way to incorporate seed information, in which the distance between entities in a positive pair (i.e., known alignment in $\mathcal{S}$) is minimized, while that referred to a negative pair is enlarged. As the most efficient choice, we consider f_a as Euclidean distance between two embeddings (Sun et al., 2020b; Guo et al., 2019; Wang et al., 2018; Pei et al., 2019a; Wu et al., 2019). The corresponding alignment loss can be written as follows:

$$\begin{aligned}\mathcal{L}_a = &\sum_{(e_i^1, e_i^2)\in\mathcal{S}} ||\mathbf{e}_i^1 - \mathbf{e}_i^2|| \\ &+ \sum_{(e_{i'}^1, e_{j'}^2)\in\mathcal{S}^-} ReLU(\alpha - ||\mathbf{e}_{i'}^1 - \mathbf{e}_{j'}^2||),\end{aligned} \tag{2}$$

where $\mathcal{S}^-$ denotes the set of negative pairs. α is the minimal margin allowed between entities in each negative entity pair. The function $ReLU$ (Nair & Hinton, 2010) will filter out negative pairs with larger margins.

On the other hand, the score function f_s also has various design choices (Sun et al., 2017; Wang et al., 2018). Most methods (Chen et al., 2017; Sun et al., 2017; Pei et al., 2019a;b; Sun et al., 2018) choose TransE as their score function, i.e., $f_s(\tau) = f_s((e_i, r, e_j)) = ||\mathbf{e}_i + \mathbf{r} - \mathbf{e}_j||, \tau = (e_i, r, e_j) \in \mathcal{T}_1 \cup \mathcal{T}_2$. The corresponding loss is:

$$\begin{aligned}\mathcal{L}_s = &\sum_{\tau\in\mathcal{T}_1\cup\mathcal{T}_2} ReLU(f_s(\tau) - \lambda) \\ &+ \sum_{\tau'\in\mathcal{T}_1^-\cup\mathcal{T}_2^-} ReLU(\lambda - f_s(\tau')),\end{aligned} \tag{3}$$

where $\mathcal{T}_1^-$ and $\mathcal{T}_2^-$ are negative triple sets. $\mathcal{L}_s$ is a margin-based loss in which the distance $d(\mathbf{e}_i + \mathbf{r}, \mathbf{e}_j)$ in a positive triple should at least be smaller than $\lambda \geq 0$, while larger than λ for negative ones.

2.2. Understanding and Rethinking EEA

We illustrate how an EEA method works with Figure 2. When the KG embedding model has a small margin, as shown in the left of Figure 2a, the entities with few neighbors can be constrained tightly. With aligned entity pairs serving as anchors, the circles are very close to each other. Therefore, two entities stay closely in the overlapped intersection areas. By contrast, there is no valid area for the entities with rich neighbors. The entities in the right of Figure 2a are not "fully expressed" (Kazemi & Poole, 2018; Trouillon et al., 2016).

If we enlarge the margin, as shown in Figure 2b, the embeddings for entities with rich neighbors can be correctly assigned. However, the intersection areas for entities with few neighbors are too loose to bound the underlying aligned entities. The two embeddings are not as similar as we expect in the vector space. We summarize the observations as:

Proposition 2.2 (Discrepancy Bound)**.** *The embedding difference of two potentially aligned entities (e_x^1, e_y^2) is bound by ϵ, which is proportional to the hyper-parameter λ:*

$$\exists \epsilon \propto \lambda, ||\mathbf{e}_x^1 - \mathbf{e}_y^2|| \leq \epsilon. \tag{4}$$

Proof. We start with the case that each entity in (e_x^1, e_y^2) has only one neighbor, connected by the same relation $r^1 = r^2$. We assume that their neighbors are actually a pair of aligned entities $(e_i^1, e_i^2) \in \mathcal{S}$. With a well-trained and almost optimal EEA model, we have $\mathbf{e}_i^1 = \mathbf{e}_i^2$ (as $\mathcal{L}_a$ is minimized) and $\mathbf{r}^1 = \mathbf{r}^2$ (denoted by $\mathbf{r}$ for simplicity). According to Equation (3), we have:

$$||f_s(\mathbf{e}_x^1, \mathbf{r}, \mathbf{e}_i^1)|| \approx ||f_s(\mathbf{e}_y^2, \mathbf{r}, \mathbf{e}_i^2)|| \leq \lambda. \tag{5}$$

Without loss of generality, we consider the score function

of TransE as f_s, and then derive:

$$||\mathbf{e}_x^1 + \mathbf{r} - \mathbf{e}_i^1|| \leq \lambda, \quad ||\mathbf{e}_y^2 + \mathbf{r} - \mathbf{e}_i^2|| \leq \lambda. \tag{6}$$

For simplicity, we use a constant C to denote $\mathbf{r} - \mathbf{e}_i^1$ and $\mathbf{r} - \mathbf{e}_i^2$, such that Equation (6) will be rewritten as

$$||\mathbf{e}_x^1 + C|| \leq \lambda, \quad ||\mathbf{e}_y^2 + C|| \leq \lambda. \tag{7}$$

Then, we get

$$\begin{aligned} 2\lambda \geq & ||\mathbf{e}_x^1 + C|| + ||\mathbf{e}_y^2 + C|| \\ \geq & ||(\mathbf{e}_x^1 + C) - (\mathbf{e}_y^2 + C)|| \\ = & ||\mathbf{e}_x^1 - \mathbf{e}_y^2||. \end{aligned} \tag{8}$$

Now, we consider a more complicated case: the neighbors of e_x^1 and e_y^2 are not in the known alignment set. We denote $\mathbf{r} - \mathbf{e}_i^1$ and $\mathbf{r} - \mathbf{e}_i^2$ by C_i^1 and C_i^2, respectively. If the neighbors of the neighbors e_i^1, e_i^2 are a pair of known alignment, we will have $||C_i^1 - C_i^2|| = ||\mathbf{e}_i^2 - \mathbf{e}_i^1|| \leq 2\lambda$, otherwise we can recursively navigate more neighbors. Therefore, we have:

$$\begin{aligned} 2\lambda \geq & ||\mathbf{e}_x^1 - \mathbf{e}_y^2 + (C_i^1 - C_i^2)|| \\ \geq & ||\mathbf{e}_x^1 - \mathbf{e}_y^2|| - ||C_i^1 - C_i^2||, \end{aligned} \tag{9}$$

which results in an looser bound:

$$||\mathbf{e}_x^1 - \mathbf{e}_y^2|| \leq 2\lambda + ||C_i^1 - C_i^2||. \tag{10}$$

For the case that the entities (e_x^1, e_y^2) have more than one neighbors, the bound will be further tighten as the embeddings are constrained by multiple triples. □

Proposition 2.2 suggests that decreasing the value of λ will decrease the embedding discrepancy in the underlying aligned entity pairs. However, previous studies (Trouillon et al., 2016; Kazemi & Poole, 2018) have proved that λ cannot be set as small as we want. This is because TransE with a small margin is not sufficient to fully capture the semantics contained in triples. Some empirical statistics (Sun et al., 2018) also illustrate such results. Enlarging the margin λ, on the other hand, will bring significant variance between $\mathbf{e}_x^1$ and $\mathbf{e}_y^2$, especially for those entities with few neighbors. For the models that do not belong to the TransE family, e.g., neural-based like ConvE (Dettmers et al., 2018) or composition-based like ComplEx (Trouillon et al., 2016), as well as hyperbolic models like (Chami et al., 2020; Sun et al., 2020a), they are more expressive than TransE. In this case, entities with sufficient neighbors can be correctly modeled, while entities with only a few neighbors are also less constrained. Therefore, those models allow more diversity between $\mathbf{e}_x^1$ and $\mathbf{e}_y^2$. We believe this is why they performed badly in the EA task (Guo et al., 2019; Sun et al., 2020c).

In short, most existing works adopt the above strategy to learn cross-KG embeddings for EA, which makes them stuck in balancing between the bound and the expressiveness. They want all given triples to be properly encoded, and meanwhile the discrepancy between potentially aligned entities to be tightly restrained. In this paper, we explore a new approach to align the conditional embedding distributions of two KGs to fulfill this goal.

3. Neural Ontology

In real-world KGs, entities conform with the rules defined by some special triples, which are known as *axioms*. Similarly, we call the entity embedding distributions *neural axioms*, and the process of aligning multiple neural axioms between two KGs *neural ontology alignment*. Aligning them allows us to regularize the entity embeddings at a high level.

3.1. Aligning Embedding Distributions with Adversarial Learning

We start from an introduction to the entity embedding distribution. It is well-known that entity embeddings can implicitly capture ontology-level information (Bordes et al., 2013; Yang et al., 2015). For example, entities from the same class are usually spatially close to each other in the vector space. In the other way around, a cluster of entity embeddings in the vector space may also indicate the existence of a class. Our goal is to exploit such ontology-level knowledge from the embedding distributions. Therefore, we define the basic neural axiom as the distribution of entity embeddings:

$$\mathbb{A}_E = p_e(\mathbf{e}), \tag{11}$$

where p_e is the entity probability distribution over the sample set E (in this case it equals to the entity set). Aligning the basic neural axioms $\mathbb{A}_E^1$ and $\mathbb{A}_E^2$ of two KGs is trivial. We take the advantages of existing domain adaptation (DA) methods (Ganin & Lempitsky, 2015; Shen et al., 2018; Ben-David et al., 2010; Courty et al., 2017) that also aims to align the feature distributions of two datasets for knowledge transferring. Specifically, we consider the method based on adversarial learning (Goodfellow et al., 2014; Arjovsky et al., 2017), where a discriminator is employed to distinguish entity embeddings of entity set $\mathcal{E}_1$ from those of $\mathcal{E}_2$ (or vice versa). The embeddings, by contrast, try to confuse the discriminator. Therefore, the same semantics in two KGs shall be encoded in the same way to fool the discriminator. The corresponding empirical Wasserstein distance based loss (Arjovsky et al., 2017; Shen et al., 2018) is:

$$\mathcal{L}_{\mathbb{A}_E} = \mathbb{E}_{\mathbb{A}_E^1}[f_w(\mathbf{e})] - \mathbb{E}_{\mathbb{A}_E^2}[f_w(\mathbf{e})], \tag{12}$$

where f_w is the learnable domain critic that maps the embedding vector to a scalar value. As suggested in (Arjovsky et al., 2017), the empirical Wasserstein distance can be approximated by maximizing $\mathcal{L}_{\mathbb{A}_E}$, if the parameterized family of f_w are all 1-Lipschitz.

(a) (b) (c) (d) (e)

Figure 3: Example of different embedding distributions. (a) Entity embedding distributions of two KGs, i.e., $\mathbb{A}_E$. $Blue$ points denote entities in $\mathcal{G}_1$, and $orange$ ones are entities in $\mathcal{G}_2$. The two embedding sets are nearly uniformly distributed and almost aligned (based on the EEA model RDGCN (Wu et al., 2019), the same below). (b) The head entity embedding distributions of relation "genre". The two distributions are only aligned partially. (c) Head entity embedding distributions conditioned on "genre", i.e., $\mathbb{A}_{E_h|r_i}$. Two conditional distributions are aligned as expected. (d) The head entity distributions conditioned on three different relations: "genre" (colors: $< blue, orange >$), "writer" (colors: $< purple, pink >$), "brithPlace" (colors: $< green, red >$). The distributions corresponding to the first two relations are overlapped, while a clear decision boundary between them and the last one is observed. (e) Triple embedding distributions conditioned on relations "artist" (colors: $< blue, orange >$) and "musicalArtist" (colors: $< purple, pink >$), respectively. The entity embeddings referred to sub-relation "musicalArtist" are covered by those corresponding to "artist".

Although the above method provides a general solution for many alignment tasks, it is not completely appropriate to the EEA problem. The most important reason is that entity embeddings are initialized randomly and tend to uniformly distributed in the vector space, as shown in Figure 3a.

Recall that the alignment loss $\mathcal{L}_a$ consists of two terms. The first is $\sum_{(e_i^1,e_i^2)\in\mathcal{S}} ||\mathbf{e}_i^1 - \mathbf{e}_i^2||$, which aims to minimize the difference of embeddings for each positive pair. The cardinality of $\mathcal{S}$ is usually small in the weakly supervised setting. However, a large size of negative samples are used for contrastive learning, which means that $||\mathcal{S}|| \ll ||\mathcal{S}^-||$. The model actually puts more effort into the second term $\sum_{(e_{i'}^1,e_{j'}^2)\in\mathcal{S}^-} ReLU(\alpha - ||\mathbf{e}_{i'}^1 - \mathbf{e}_{j'}^2||)$, of which the main target is to randomly push the embeddings of non-aligned entities away from each other. On the other hand, $\mathcal{L}_s$ is also a contrastive loss and has a similar effect on maximizing the pairwise distance between a positive entity and its sampled negative ones. Therefore, we may conclude that:

Proposition 3.1 (Uniformity)**.** *The entity embeddings of two KGs tend to be uniformly distributed in the vector space as an EEA model is optimized.*

This characteristic has also been studied in (Wang & Isola, 2020) in other representation learning problems. The uniformity property reveals that the information of entities are efficiently encoded to maximize the entropy. However, for the given EEA problem, the entity embedding distributions of two KGs will be similar to each other, especially when the seed alignment pairs exist. Thus, only aligning the basic neural axioms is less helpful to facilitate EEA.

3.2. Conditional Neural Axiom

We can estimate the conditional distributions rather than the raw distributions to avoid the problem brought from the uniformity property. Specifically, we name *conditional neural axioms* to describe the entity (or triple) embedding distributions under specific semantics. For example, the head entity embedding distribution conditioned on the relation embedding $\mathbf{r}_i$ can be defined as:

$$\mathbb{A}_{E_h|r_i} = p_{e_h|r_i}(\mathbf{e}_h|\mathbf{r}_i), \tag{13}$$

where $p_{e_h|r_i}$ is the conditional probability distribution of the head entities given r_i. The corresponding sample set is defined as:

$$E_h|r_i = \{e \,|\, \exists e', (e, r_i, e') \in \mathcal{T}\}. \tag{14}$$

Following the similar rule, we can define the conditional triple distribution $\mathbb{A}_{E_{h,t}|r_i}$ with the sample set

$$E_{h,t}|r_i = \{(e_h, e_t) \,|\, (e_h, r_i, e_t) \in \mathcal{T}\}. \tag{15}$$

Numerous methods have been proposed to process the neural conditioning operation, ranging from addition and concatenation (Mirza & Osindero, 2014; Wang et al., 2014; Dettmers et al., 2018), to matrix multiplication (Lin et al., 2015a; Ji et al., 2015; Nguyen et al., 2016). Rather than developing new methods, we value more on its common merit: projecting the entities to a relation-specific subspace (Wang et al., 2014; Lin et al., 2015a; Nguyen et al., 2016). Hence, the corresponding embedding distributions conditioned on different relations become discriminative, compared to uniformly distributed in the original embedding space.

Furthermore, conditional neural axioms capture high-level ontology knowledge:

Proposition 3.2 (Expressiveness)**.** *Aligning the conditional neural axioms minimizes the embedding discrepancy of two KGs at the ontology level, without the need of type/class information.*

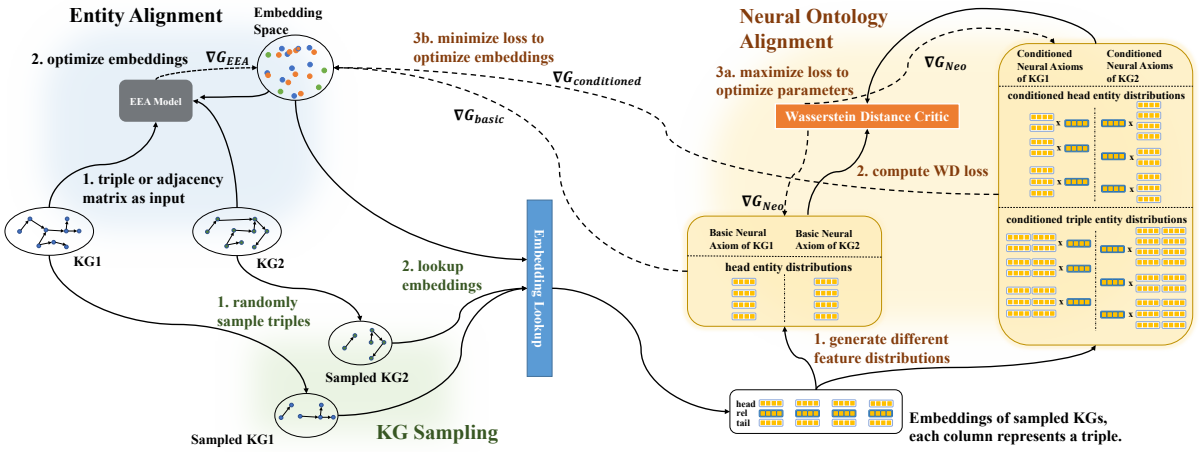


Figure 4: Architecture of NeoEA. Solid lines denote forward propagation, while dotted lines represent backward propagation. The architecture consists of three decoupled modules: entity alignment, KG sampling, and neural ontology alignment.

Proof. See Appendix A for details. □

We take $\mathbb{A}_{E_h|r_i}$ as an example that summarizes the empirical "Object Property Domain" axiom of r_i in OWL 2 EL (Baader et al., 2005). Supposed there exists such an axiom stating that the head entities of r_i should belong to some specific class c (e.g., only head entities belonging to the class "Person" have the relation "birthPlace"). We further suppose that there exists a classifier $f_c(\mathbf{e}) \in [0, 1]$, such that $f_c(\mathbf{e}_j) = 1$ if head entity e_j belongs to class c, and 0 otherwise. Then, with the knowledge of the given axiom, one may derive the following rule:

$$\forall e \in \{e \,|\, \exists e', (e, r_i, e') \in \mathcal{T}_1 \cup \mathcal{T}_2\},\ f_c(\mathbf{e}) = 1, \quad (16)$$

which is equivalent to:

$$\mathbb{E}_{\mathbb{A}^1_{E_h|r_i}}[f_c(\mathbf{e})] = \mathbb{E}_{\mathbb{A}^2_{E_h|r_i}}[f_c(\mathbf{e})] = 1, \quad (17)$$

both of which means that all head entities of r_i in either KG should be correctly classified to c. Then, we have:

$$\mathbb{E}_{\mathbb{A}^1_{E_h|r_i}}[f_c(\mathbf{e})] - \mathbb{E}_{\mathbb{A}^2_{E_h|r_i}}[f_c(\mathbf{e})] = 0. \quad (18)$$

In fact, we do not have such knowledge about r_i and class c. Instead, we can leverage a neural function $f_{c'}(\mathbf{e}|\mathbf{r}_i)$ to estimate f_c empirically. In this way, $\mathbb{A}^1_{E_h|r_i}$ and $\mathbb{A}^2_{E_h|r_i}$ are supposed to be aligned to minimize the loss corresponding to the above rule. Therefore, we deduce this problem back to a similar form to Equation 12, i.e.,

$$\mathcal{L}_{\mathbb{A}_{E_h|r_i}} = \mathbb{E}_{\mathbb{A}^1_{E_h|r_i}}[f_{c'}(\mathbf{e}\,|\,\mathbf{r}_i)] - \mathbb{E}_{\mathbb{A}^2_{E_h\,|\,r_i}}[f_{c'}(\mathbf{e}|\mathbf{r}_i)], \quad (19)$$

which suggests that aligning the above conditional neural axioms can minimize the discrepancy of potential "Object Property Domain" axioms between two KGs.

Example 1 (OWL2 axiom: ObjectPropertyDomain). As shown in Figures 3b and 3c, we assume that the head entities of relation "genre" are under the class "Work of Art" (although it does not exist). It is clear that the head entity embedding distributions are only partially aligned in Figure 3b, while those in Figure 3c are matched well.

In Figure 3d, we present a more complicated example. The head entities of relations "genre" and "writer" mainly belong to "Work of Art", which show overlapped distributions (blue-orange, pink-purple). By contrast, there exists a clear decision boundary between them and the distributions conditioned on relation "birthPlace" (red-green), as the head entities of relation "birthPlace" are under the class "Person".

Example 2 (OWL2 axiom: SubObjectPropertyOf). We consider two relations, "musicalArtist" and "artist" as an example, where the former is the latter's sub-relation. In Figure 3e, the triple distributions conditioned on "musicalArtist" (colors: pink-purple) are covered by those conditioned on "artist" (colors: orange-blue).

3.3. Implementation

We illustrate the overall structure in Figure 4. The framework can be divided into three modules:

Entity Alignment This module aims at encoding the semantics of KGs into embeddings. Almost all existing EEA models can be used here, no matter what the input data look like (e.g., triples or adjacency matrices).

KG Sampling For each KG, we randomly sample a sub-KG to estimate the data distributions of neural axioms. It is more efficient than separately sampling candidates for each axiom, especially when KGs get big.

Table 1: Results on V1 datasets (5-fold cross-validation).

Models	EN-FR			EN-DE			D-W			D-Y		
	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR
BootEA (Sun et al., 2018)	.507	.718	.603	.675	**.820**	**.740**	.572	.744	.649	.739	.849	.788
BootEA + NeoEA	**.521**	**.733**	**.617**	**.676**	**.820**	**.740**	**.579**	**.753**	**.658**	**.756**	**.859**	**.797**
SEA (Pei et al., 2019a)	.280	.530	.397	.530	.718	.617	.360	.572	.458	.500	.706	.591
SEA + NeoEA	**.320**	**.584**	**.443**	**.586**	**.766**	**.668**	**.389**	**.608**	**.490**	**.549**	**.752**	**.638**
RSN (Guo et al., 2019)	.393	.595	.487	.587	.752	.662	.441	.615	.521	.514	.655	.580
RSN + NeoEA	**.399**	**.597**	**.490**	**.600**	**.759**	**.673**	**.450**	**.624**	**.530**	**.522**	**.663**	**.588**
RDGCN (Wu et al., 2019)	.755	.854	.800	.830	.895	.859	.515	.669	.584	.931	.969	.949
RDGCN + NeoEA	**.775**	**.868**	**.817**	**.846**	**.908**	**.874**	**.527**	**.671**	**.592**	**.941**	**.972**	**.955**

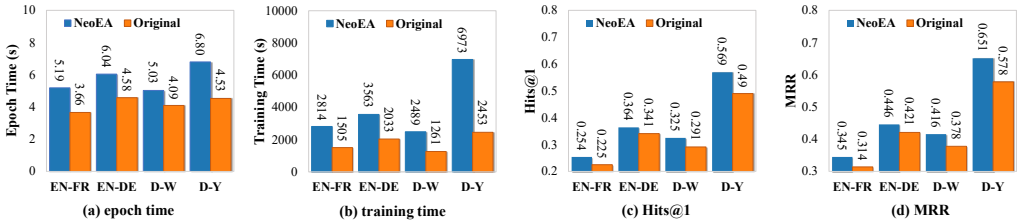


Figure 5: Results on OpenEA 100K datasets, with the fastest EEA model SEA as baseline.

Algorithm 1 NeoEA

Input: two KGs $\mathcal{G}_1, \mathcal{G}_2$, the seed alignment set $\mathcal{S}$, the EEA model $\mathcal{M}(f_s, f_a)$, number of steps for NeoEA n;
Initialize all variables;
repeat
  for $i := 1$ **to** n **do**
    Sample sub-KGs from respective KGs $\mathcal{G}_1, \mathcal{G}_2$;
    Compute the Wasserstein-distance-based loss $\mathcal{L}_w$ for each pair of neural axioms;
    Optimize f_w by maximizing $\mathcal{L}_w$.
  end for
  Sample sub-KGs from respective KGs $\mathcal{G}_1, \mathcal{G}_2$;
  Compute $\mathcal{L}_w$ for each pair of neural axioms;
  Compute the losses $\mathcal{L}_a, \mathcal{L}_s$ of the EEA model $\mathcal{M}$;
  Minimize $\mathcal{L}_a, \mathcal{L}_s, \mathcal{L}_w$;
until the alignment loss on the validation set converges.

Neural Ontology Alignment As aforementioned, for each pair of embedding distributions, we align them by minimizing the empirical Wasserstein distance and optimizing by gradient ascent/descent.

We present the algorithm in Algorithm 1. For the detailed implementation, please see Appendix B.

4. Experiments

In this section, we empirically verify the effectiveness of NeoEA by a series of experiments[1].

[1]https://github.com/guolingbing/NeoEA

4.1. Settings

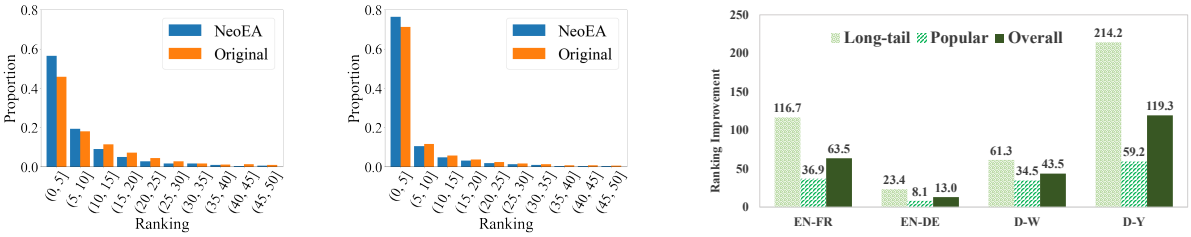
We selected four best-performing and representative models as our baselines: BootEA (Sun et al., 2018), a TransE-based EEA model with only structure data (i.e., triples) as input; SEA (Pei et al., 2019a), a TransE-based model with both structure and entity attribute data as input; RSN (Guo et al., 2019), an RNN-based EEA model with only structure data as input; RDGCN (Wu et al., 2019), a GCN-based model with both structure and attribute data as input. The whole framework is based on OpenEA (Sun et al., 2020c). We modified only the initialization of the original project and kept the optimal hyper-parameter settings in OpenEA to ensure a fair comparison. We used the latest benchmark provided by OpenEA (Sun et al., 2020c), which consists of four sub-datasets with two density settings. Specifically, "D-W", "D-Y" denote "DBpedia (Auer et al., 2007)-WikiData (Vrandečić & Krötzsch, 2014)", "DBpedia-YAGO (Fabian et al., 2007)", respectively. "EN-DE" and "EN-FR' denote two cross-lingual datasets, both of which are sampled from DBpedia. The entity degree distributions in "V1" datasets are similar to those in the original KGs, while the average degree in "V2" datasets are doubled.

4.2. Empirical Comparisons

The main results on V1 datasets are shown in Table 1. Although the performance of four baseline models varied from different datasets, all of them gained improvement with NeoEA. This demonstrates that aligning the neural ontology is beneficial for all four kinds of EEA models. Furthermore, we find the performance improvement on SEA and RDGCN was more significant than that on other two methods, as

Table 2: Results of ablation study based on the best-performing model RDGCN, on V1 datasets.

Models	EN-FR			EN-DE			D-W			D-Y		
	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR
Full	**.775**	**.868**	**.817**	**.846**	**.908**	**.874**	**.527**	**.671**	**.592**	**.941**	**.972**	**.955**
Partial	.771	.863	.813	.840	.900	.871	.523	.669	.590	.936	.971	.952
Basic	.755	.853	.799	.827	.895	.858	.512	.656	.578	.931	.969	.948
Original	.755	.854	.800	.830	.895	.859	.515	.669	.584	.931	.969	.949



(a) long-tail entities (EN-FR) (b) popular entities (EN-FR) (c) average ranking improvement on four datasets

Figure 6: A comparison of ranking results between NeoEA and the baseline method SEA. The ranking results on the testing datasets are grouped by the ranking intervals (Figures 6a, 6b) and entity types (Figure 6c), respectively.

BootEA and RSN are not typical EEA models. BootEA has a sophisticated bootstrapping procedure, which may be challenging to be injected with NeoEA. RSN tries to capture long-term dependencies. The complicated objective may conflict with NeoEA more or less. Even though we still observe relatively significant improvement on some datasets (e.g., EN-DE and D-Y). Therefore, we believe their performance can be refined through a joint hyper-parameter turning with NeoEA, which we leave to future work. The results on V2 datasets (i.e., the denser and simpler ones) are presented in Appendix C. Briefly, the improvement is relatively smaller than that on V1 datasets but NeoEA still outperformed all the baselines.

4.3. The Scalability and Efficiency of NeoEA

We also conducted experiment on the OpenEA 100K datasets to evaluate performance of NeoEA on larger KGs. We used a single TITAN RTX for training, and SEA (the fastest model) as the basic EEA model. In theory, NeoEA does not have multiple GCN/GAT layers nor the pair-wise similarity estimation on whole graphs. The embedding distributions are also obtained from the sampled KG. Therefore, NeoEA is applicable to large-scale datasets.

From Figure 5, we can find that the time for training one epoch (Figure 5a) was not evidently increased, especially considering the cost of switching the optimizers. On the other hand, the overall training time was nearly doubled (Figure 5b), caused by the adversarial training procedure. Even though the loss converged more slowly in NeoEA, we should notice that the overall training time was much less than that of complicated EEA models, e.g., BootEA (35,000+ seconds). In Figures 5c and 5d, we can find the improvement for Hits@1 and MRR was significant, especially on the D-Y dataset that took longest training time.

4.4. The Necessity of Conditional Neural Axioms

We designed an experiment to verify some claims in Section 3. We choose the best-performing model RDGCN as our baseline. In Table 2, "Full" denotes NeoEA with the full set of neural axioms. "Partial" denotes NeoEA without the conditional triple axioms. We removed the conditional entity axioms from "Partial" to construct "Basic", and the last one, "Original", denotes the original EEA model.

Aligning the basic axiom was less effective or even harmful to the model. This result empirically demonstrates our assumption that the uniformity property of the learned entity embeddings will make the embedding distribution alignment meaningless. On the other hand, aligning only a part of conditional axioms $\mathbb{A}_{E_h|r_i}, \mathbb{A}_{E_t|r_i}$ that describe entity embedding distributions conditioned on relation embeddings was significantly helpful for the model. Also, the additional improvement was observed with the full conditional axioms.

4.5. Further Analysis of the Bound

We have shown the embedding discrepancy between each underlying aligned pair is bounded by ϵ in Section 2. This section provides empirical statistics to verify this point. We manually split the entities into two groups: (1) long-tail entities, which have at most two neighbors that do not belong to known aligned pairs; (2) popular entities, the remaining.

We draw the histograms of alignment rankings w.r.t. respective groups based on the EEA model SEA. From Figures 6a, 6b, we can find the proportion of the inexact alignments (i.e., ranking > 5) for long-tail entities is evidently larger than that of popular entities, especially for the bins $(5, 20]$. This observation verified that the long-tail entities are less constrained compared to those popular entities in EEA problem. Furthermore, with NeoEA, the rankings of those long-tail entities were improved more significantly than those of popular entities, which demonstrates that NeoEA indeed tightened the representation discrepancy of those less restrained entities. We report the average ranking improvement on four V1 datasets in Figure 6c, which shows consistent results. It is worth noting that the ranking improvement for long-tail entities is more than twice as larger as that for popular entities, except the D-W dataset.

5. Conclusion and Future Work

In this paper, we proposed a new approach to learn KG embeddings for entity alignment. We proved its expressiveness theoretically and demonstrated its efficiency on extensive experiments. Four state-of-the-art EEA methods gained evident benefits with NeoEA, where the conditional neural axiom is the key component. For the future work, we plan to study how to extend NeoEA with realistic ontology knowledge for further improvement.

Acknowledgements

We want to thank all anonymous reviewers for their invaluable comments. This work is funded by NSFCU19B2027/NSFC91846204, National Key R&D Program of China (Funding No.SQ2018YFC000004), and Zhejiang Provincial Natural Science Foundation of China (No.LGG22F030011). Zequn Sun's work was supported by Program A for Outstanding PhD Candidates of Nanjing University.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. *CoRR*, 2017.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. G. Dbpedia: A nucleus for a web of open data. In *ISWC*, pp. 722–735, 2007.

Baader, F., Brandt, S., and Lutz, C. Pushing the EL envelope. In *IJCAI*, pp. 364–369, 2005.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Mach. Learn.*, 79:151–175, 2010.

Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *NIPS*, pp. 2787–2795, 2013.

Chami, I., Wolf, A., Juan, D., Sala, F., Ravi, S., and Ré, C. Low-dimensional hyperbolic knowledge graph embeddings. In *ACL*, pp. 6901–6914, 2020.

Chen, J., Hu, P., Jiménez-Ruiz, E., Holter, O. M., Antonyrajah, D., and Horrocks, I. Owl2vec*: embedding of OWL ontologies. *Mach. Learn.*, 110(7):1813–1845, 2021.

Chen, M., Tian, Y., Yang, M., and Zaniolo, C. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*, pp. 1511–1517, 2017.

Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. Joint distribution optimal transportation for domain adaptation. In *NeurIPS*, pp. 3730–3739, 2017.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. Convolutional 2D knowledge graph embeddings. In *AAAI*, pp. 1811–1818, 2018.

El-Roby, A. and Aboulnaga, A. ALEX: Automatic link exploration in linked data. In *SIGMOD*, pp. 1839–1853, Melbourne, Australia, 2015.

Fabian, M., Gjergji, K., Gerhard, W., et al. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *WWW*, pp. 697–706, 2007.

Fey, M., Lenssen, J. E., Morris, C., Masci, J., and Kriege, N. M. Deep graph matching consensus. In *ICLR*, 2020. URL https://openreview.net/forum?id=HyeJf1HKvS.

Ganin, Y. and Lempitsky, V. S. Unsupervised domain adaptation by backpropagation. In *ICML*, pp. 1180–1189, 2015.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks, 2014.

Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P. F., and Sattler, U. OWL 2: The next step for OWL. *J. Web Semant.*, 6(4):309–322, 2008.

Guo, L., Sun, Z., and Hu, W. Learning to exploit long-term relational dependencies in knowledge graphs. In *ICML*, pp. 2505–2514, 2019.

Horrocks, I., Kutz, O., and Sattler, U. The even more irresistible SROIQ. In Doherty, P., Mylopoulos, J., and Welty, C. A. (eds.), *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pp. 57–67. AAAI Press, 2006. URL http://www.aaai.org/Library/KR/2006/kr06-009.php.

Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In *ACL*, pp. 687–696, 2015.

Kazemi, S. M. and Poole, D. Simple embedding for link prediction in knowledge graphs. In *NeurIPS*, pp. 4289–4300, Montréal, Canada, 2018.

Lacoste-Julien, S., Palla, K., Davies, A., Kasneci, G., Graepel, T., and Ghahramani, Z. SiGMa: Simple greedy matching for aligning large knowledge bases. In *KDD*, pp. 572–580, Chicago, IL, USA, 2013.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pp. 2181–2187, 2015a.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *CoRR*, 2014.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *ICML*, pp. 807–814, 2010.

Nguyen, D. Q., Sirts, K., Qu, L., and Johnson, M. STransE: A novel embedding model of entities and relationships in knowledge bases. In *NAACL*, pp. 460–466, San Diego, USA, 2016.

Pei, S., Yu, L., Hoehndorf, R., and Zhang, X. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *WWW*, pp. 3130–3136, 2019a.

Pei, S., Yu, L., and Zhang, X. Improving cross-lingual entity alignment via optimal transport. In *IJCAI*, pp. 3231–3237, 2019b.

Shen, J., Qu, Y., Zhang, W., and Yu, Y. Wasserstein distance guided representation learning for domain adaptation. In *AAAI*, pp. 4058–4065, 2018.

Suchanek, F. M., Abiteboul, S., and Senellart, P. PARIS: Probabilistic alignment of relations, instances, and schema. *PVLDB*, 5:157–168, 2012.

Sun, Z., Hu, W., and Li, C. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*, pp. 628–644, 2017.

Sun, Z., Hu, W., Zhang, Q., and Qu, Y. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, pp. 4396–4402, 2018.

Sun, Z., Chen, M., Hu, W., Wang, C., Dai, J., and Zhang, W. Knowledge association with hyperbolic knowledge graph embeddings. In *EMNLP*, pp. 5704–5716, 2020a.

Sun, Z., Wang, C., Hu, W., Chen, M., Dai, J., Zhang, W., and Qu, Y. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *AAAI*, pp. 222–229, 2020b.

Sun, Z., Zhang, Q., Hu, W., Wang, C., Chen, M., Akrami, F., and Li, C. A benchmarking study of embedding-based entity alignment for knowledge graphs. *CoRR*, abs/2003.07743, 2020c.

Trouillon, T., Welbl, J., Riedel, S., Éric Gaussier, and Bouchard, G. Complex embeddings for simple link prediction. In *ICML*, pp. 2071–2080, 2016.

Vrandečić, D. and Krötzsch, M. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57: 78–85, 2014.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pp. 9929–9939, 2020.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pp. 1112–1119, 2014.

Wang, Z., Lv, Q., Lan, X., and Zhang, Y. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*, pp. 349–357, 2018.

Wu, Y., Liu, X., Feng, Y., Wang, Z., Yan, R., and Zhao, D. Relation-aware entity alignment for heterogeneous knowledge graphs. In *IJCAI*, pp. 5278–5284, 2019.

Yang, B., Yih, W., He, X., Gao, J., and Deng, L. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, 2015. URL http://arxiv.org/abs/1412.6575.

Ye, R., Li, X., Fang, Y., Zang, H., and Wang, M. A vectorized relational graph convolutional network for multi-relational network alignment. In *IJCAI*, pp. 4135–4141, 2019.

Zhu, H., Xie, R., Liu, Z., and Sun, M. Iterative entity alignment via joint knowledge embeddings. In *IJCAI*, pp. 4258–4264, 2017.

A. Proof for Proposition 3.2

Aligning the conditional neural axioms minimizes the embedding discrepancy of two KGs at the ontology level.

Proof. We split the proofs according to the types of axioms in OWL2 EL (Baader et al., 2005):

ObjectPropertyDomain, ObjectPropertyRange. The proof for ObjectPropertyDomain has been presented in Section 3, and that for ObjectPropertyRange is similar.

Table 3: Results on V2 datasets (5-fold cross-validation).

Models	EN-FR			EN-DE			D-W			D-Y		
	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR
RSN (Guo et al., 2019)	.579	.759	.662	.791	.890	.837	.723	.854	.782	.933	.974	.951
RSN + NeoEA	**.583**	**.760**	**.666**	**.794**	**.892**	**.839**	**.729**	**.858**	**.787**	**.935**	**.976**	**.953**
SEA (Pei et al., 2019a)	.360	.651	.494	.606	.779	.687	.567	.770	.660	.899	.950	.923
SEA + NeoEA	**.375**	**.666**	**.508**	**.637**	**.800**	**.712**	**.588**	**.784**	**.677**	**.917**	**.959**	**.936**
BootEA (Sun et al., 2018)	.660	.850	.745	.833	.912	.869	.821	**.926**	.867	**.958**	**.984**	**.969**
BootEA + NeoEA	**.665**	**.853**	**.749**	**.834**	**.916**	**.870**	**.822**	**.926**	**.869**	**.958**	**.984**	**.969**
RDGCN (Wu et al., 2019)	.847	.919	.880	.833	.891	.860	.623	.757	.684	.936	.966	.950
RDGCN + NeoEA	**.864**	**.933**	**.896**	**.849**	**.902**	**.874**	**.632**	**.760**	**.690**	**.940**	**.970**	**.953**

ReflexiveObjectProperty, IrreflexiveObjectProperty. If we say that a relation r_i is reflexive, it must satisfy

$$\forall(e, r_i, e') \in \mathcal{T},\ (e, r_i, e) \in \mathcal{T}, \tag{20}$$

which means each head entities of r_i must be connected by r_i to itself. The above rule suggests that we can align the underlying *reflexive* knowledge by minimizing the discrepancy between triple distributions conditioned on relation r_i, i.e., aligning $\mathbb{A}^1_{E_{(h,t)}|r_i}$ with $\mathbb{A}^2_{E_{(h,t)}|r_i}$. The similar to IrreflexiveObjectProperty axiom.

FunctionalObjectProperty, InverseFunctionalObjectProperty. We first introduce FunctionalObjectProperty axiom. It compels each head entity e connected by relation r_i to have exactly one tail entity, implying the following rule:

$$\forall(e, r_i, e') \in \mathcal{T}, \forall e'' \in \mathcal{E},\ (e, r_i, e'') \notin \mathcal{T}. \tag{21}$$

The above rule is also related to the triple distribution conditioned on r_i. The similar to the InverseFunctionalObjectProperty axiom.

SymmetricObjectProperty, AsymmetricObjectProperty. The first axiom can state a relation r_i is symmetric, that is,

$$\forall(e, r_i, e') \in \mathcal{T},\ (e', r_i, e) \in \mathcal{T}. \tag{22}$$

It is also related to the triple distributions referred to r_i, implying that aligning $\mathbb{A}^1_{E_{(h,t)}|r_i}$ with $\mathbb{A}^2_{E_{(h,t)}|r_i}$ is sufficient to minimize the difference. The similar to the AsymmetricObjectProperty axiom.

SubObjectPropertyOf, EquivalentObjectProperties, DisjointObjectProperties and InverseObjectProperties. We show that these axioms also define rules related to triple distributions conditioned on relations. We start from SubObjectPropertyOf, which can state that relation r_i is a sub-property of relation r_j (e.g., "hasDog" is one of the sub-properties of "hasPet"). We formulate it as

$$\forall(e, r_i, e') \in \mathcal{T},\ (e', r_j, e) \in \mathcal{T}. \tag{23}$$

To align the potential SubObjectPropertyOf axioms between two KGs, we can respectively align $(\mathbb{A}^1_{E_{(h,t)}|r_i}, \mathbb{A}^2_{E_{(h,t)}|r_i})$ and $(\mathbb{A}^1_{E_{(h,t)}|r_j}, \mathbb{A}^2_{E_{(h,t)}|r_j})$, such that the joint one $(\mathbb{A}^1_{E_{(h,t)}|r_i,r_j}, \mathbb{A}^2_{E_{(h,t)}|r_i,r_j})$ will also be aligned.

Similarly, if r_i and r_j are equivalent, we can interpret the axiom as

$$\forall(e, r_i, e') \in \mathcal{T},\ (e', r_j, e) \in \mathcal{T};\quad \forall(e, r_j, e') \in \mathcal{T},\ (e', r_i, e) \in \mathcal{T}. \tag{24}$$

If they are disjoint, the corresponding rule will be

$$\forall(e, r_i, e') \in \mathcal{T},\ (e, r_j, e') \notin \mathcal{T}. \tag{25}$$

If they are inverse to each other, the rule is

$$\forall(e, r_i, e') \in \mathcal{T},\ (e', r_j, e) \in \mathcal{T}. \tag{26}$$

TransitiveObjectProperty. We show that this axiom is also related to triple distributions conditioned on r_i. Supposed that a relation r_i is transitive, then one can derive the following rule:

$$\forall(e, r_i, e') \in \mathcal{T} \& (e', r_i, e'') \in \mathcal{T},\ (e, r_i, e'') \in \mathcal{T}, \tag{27}$$

which means we can align the potential TransitiveObjectProperty axioms via minimizing the distribution discrepancy between $\mathbb{A}^1_{E_{(h,t)}|r_i}$ and $\mathbb{A}^2_{E_{(h,t)}|r_i}$. □

B. Implementation Details

For efficiency, we share the parameters of Wasserstein distance critic in different *type* (e.g, conditional/unconditional, head/triple) of neural axioms, which reduces the number of model parameters and avoids the situation that some relations only have a small number of triples. This also allows us to perform fast mini-batch training by aligning the axioms of the same type in one operation. Given the sample KGs

$\mathcal{G}'_1 = (\mathcal{E}'_1, \mathcal{R}'_1, \mathcal{T}'_1)$, $\mathcal{G}'_2 = (\mathcal{E}'_2, \mathcal{R}'_2, \mathcal{T}'_2)$, the corresponding batch loss is:

$$\begin{aligned}\mathcal{L}_{sep} = \mathcal{L}_{\mathbb{A}_{E'}} + (& \sum_{r' \in \mathcal{R}'_1} \mathbb{E}_{\mathbb{A}_{E'_h|r'}}[f_{h|r}(\mathbf{e}|\mathbf{r}')] \\ & - \sum_{r' \in \mathcal{R}'_2} \mathbb{E}_{\mathbb{A}_{E'_h|r'}}[f_{h|r}(\mathbf{e}|\mathbf{r}')]) \\ + (& \sum_{r' \in \mathcal{R}'_1} \mathbb{E}_{\mathbb{A}_{E'_{h,t}|r'}}[f_{h,t|r}(\mathbf{e}_h, \mathbf{e}_t|\mathbf{r}')] \\ & - \sum_{r' \in \mathcal{R}'_2} \mathbb{E}_{\mathbb{A}_{E'_{h,t}|r'}}[f_{h,t|r}(\mathbf{e}_h, \mathbf{e}_t|\mathbf{r}')]), \end{aligned} \quad (28)$$

where $\mathcal{L}_{\mathbb{A}_{E'}}$ is the basic axiom loss under the sampled KGs. $f_{h|r}$ and $f_{h,t|r}$ are the critic functions of two types of neural axioms, respectively. The loss $\mathcal{L}_{batch}$ will approximate to that in pairwise calculation when batch-size is considerably greater than the number of relations. We take the second term in Equation (28) as an example. For pairwise estimation, the loss should be:

$$\begin{aligned} & \sum\nolimits_{(r_1,r_2)\in\mathcal{S}_r} (\mathbb{E}_{\mathbb{A}_{E'_h|r_1}}[f_{h|r}(\mathbf{e}|\mathbf{r}_1)] - \mathbb{E}_{\mathbb{A}_{E'_h|r_2}}[f_{h|r}(\mathbf{e}|\mathbf{r}_2)]) \\ & = \sum_{(r_1,r_2)\in\mathcal{S}_r} \mathbb{E}_{\mathbb{A}_{E'_h|r_1}}[f_{h|r}(\mathbf{e}|\mathbf{r}_1)] \\ & - \sum_{(r_1,r_2)\in\mathcal{S}_r} \mathbb{E}_{\mathbb{A}_{E'_h|r_2}}[f_{h|r}(\mathbf{e}|\mathbf{r}_2)], \end{aligned} \quad (29)$$

where $\mathcal{S}_r \subset \mathcal{R}_1 \times \mathcal{R}_2$ denotes the set of all aligned relation pairs. The above equation suggests that the pairwise loss is based on the respective relation sets of two KGs, not constrained by each pair of aligned relations. Generally, the number of relations is much smaller than the number of sampled triples in one batch, which means that $\mathcal{R}'_1, \mathcal{R}'_2$ in Equation (28) can cover a large proportion of elements in the full relation sets $\mathcal{R}_1, \mathcal{R}_2$. Therefore, we used $\mathcal{L}_{batch}$ to approximate the pairwise loss in the implementation.

C. Results on V2 Datasets

The results on V2 datasets are shown in Table 3. Although entities have doubled number of neighbors in V2 datasets, all baseline models still gained significant improvement with NeoEA.