Master Thesis

Inter-domain Multi-relational Link Prediction

Supervisor: Professor Hisashi Kashima

Department of Intelligence Science and Technology
Graduate School of Informatics
Kyoto University

Luu Huu Phuc

February 7, 2022

Inter-domain Multi-relational Link Prediction

Luu Huu Phuc

Abstract

Multi-relational graph is a ubiquitous and important data structure, allowing flexible representation of multiple types of interactions and relations between entities. Similar to other graph-structured data, link prediction is one of the most important tasks on multi-relational graphs and is often used for knowledge completion. When related graphs coexist, it is of great benefit to build a larger graph via integrating the smaller ones. The integration requires predicting hidden relational connections between entities belonged to different graphs (inter-domain link prediction). However, this poses a real challenge to existing methods that are exclusively designed for link prediction between entities of the same graph only (intra-domain link prediction). In this study, we propose a new approach to tackle the inter-domain link prediction problem by softly aligning the entity distributions between different domains with optimal transport and maximum mean discrepancy regularizers. Experiments on real-world datasets show that optimal transport regularizer is beneficial and considerably improves the performance of baseline methods.

複数ドメイン・複数種の関係グラフ上のリンク予測法

リュウ・ユウ・フク

内容梗概

なし。

Inter-domain Multi-relational Link Prediction

Contents

1 Introduction

Multi-relational data represents knowledge about the world and provides a graph-like structure of this knowledge. It is defined by a set of entities and a set of predicates between these entities. The entities can be objects, events, or abstract concepts while the predicates represent relationships involving two entities. A multi-relational data contains a set of facts represented as triplets (e_h, r, e_t) denoting the existence of a predicate r from subject entity e_h to object entity e_t. In a sense, multi-relational data can also be seen as a directed graph with multiple types of links (multi-relational graph).

A multi-relational graph is often very sparse with only a small subset of true facts being observed. Link prediction aims to complete a multi-relational graph by predicting new hidden true facts based on the existing ones. Many existing methods follow an embedding-based approach which has been proved to be effective for multi-relational graph completion. These methods all aim to find reasonable embedding representations for each entity (node) and each predicate (type of link). In order to predict if a fact (e_h, r, e_t) holds true, they use a scoring function whose inputs are embeddings of the entities e_h, e_t and the predicate r to compute a prediction score.

Despite achieving state of the art for link prediction tasks, existing methods are exclusively designed and limited to intra-domain link prediction. They only consider the case in which both entities belong to the same relational graph (intra-domain). When the needs for predicting hidden facts between entities of different but related graphs (inter-domain) arise, unfortunately, the existing methods are inapplicable. One of such examples is when it is necessary to build a large relational graph by integrating several existing smaller graphs whose entity sets are related. This study proposes to tackle the inter-domain link prediction problem by learning suitable embedding representation that minimizes dissimilarity between entity distributions of related domains.

To measure dissimilarity of entity distributions, two popular divergences, namely optimal transport's Wasserstein distance (WD) and the maximum mean discrepancy (MMD), are investigated. Given two probability distributions, optimal transport computes an optimal transport plan that gives the minimum total transport cost to relocate masses between the distributions. The minimum total transport cost is often known under the name of Wasserstein distance. In a sense, the computed optimal transport plan and the corresponding Wasserstein distance provide a reasonable alignment and quantity for measuring the dissimilarity between the supports/domains of the two distributions. Minimizing Wasserstein distance has been proved to be effective in enforcing the alignment of corresponding entities across different domains and is successfully applied in graph matching [1], cross-domain alignment [2], and multiple-graph link prediction problems [3]. As another popular statistical divergence between distributions, MMD computes dissimilarity by comparing the kernel mean embeddings of two distributions in a reproducing kernel Hilbert space (RKHS). It has been widely applied in two-sample tests for differentiating distributions [4, 5] and distribution matching in domain adaptation tasks [6], to name a few.

In this thesis, we formalize the inter-domain link prediction problem under a setting of two multi-relational graphs whose entities are assumed to follow the same underlying distribution. This assumption is fundamental for the proposed method to be effective in connecting entity distributions of the two graphs.

1.1 Summary of the contributions

In summary, the main contributions of the thesis are the following:

- We are the first to consider the inter-domain link prediction problem which aims to predict links between entities of different domains. We give a formal definition of the problem setting.
- We propose a method to tackle the problem and investigate two variants based on optimal transport and MMD regularizers.
- We conduct experiments on four datasets, which demonstrates promising results. The proposed method based on optimal transport regularizer outperforms baseline methods for the inter-domain link prediction tasks

while preserving similar performance for the intra-domain link prediction tasks.

1.2 Organization of the thesis

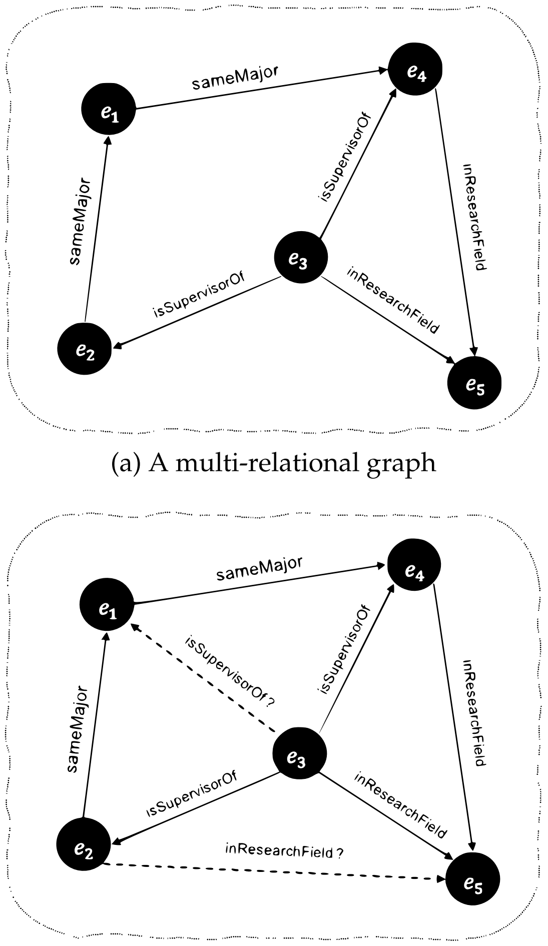
The thesis is organized as follows. Section 2 gives a general introduction of link prediction problems in multi-relational graphs and a formal problem setting of inter-domain link prediction. Existing methods for intra-domain link prediction and related works on inter-domain link prediction are reviewed in section 3. Section 4 introduces preliminary knowledge of components employed in the proposed method, which includes methods to learn embedding representations of multi-relational graphs and approximating methods to compute distributional divergences. The proposed method is elaborated in section 5. The last two sections 6 and 7 are for experiments and concluding remarks.

2 Background and Problem Setting

2.1 Link prediction in multi-relational graph

Multi-relational graph is a generalized notion of graph that allows different types of links between nodes. A multi-relational graph $G = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ consists of three components: a set of entities (nodes) $\mathcal{E}$, a set of predicates (types of links) $\mathcal{R}$, and a set of true facts (links) $\mathcal{T}$. Formally,

- $\mathcal{E} = \{e_1, ..., e_n\}$: e_i are entities. The entities could be anything, from a person, a place, or an object. E.g. $Berlin$, $Germany$, $Prof\_X$, $Stud\_A$, $Comp\_Science$, etc.
- $\mathcal{R} = \{r_1, ..., r_m\}$: r_k are predicates (relations). They classify types of relationships occuring between entities. E.g. $Is\_Capital\_Of$, $Is\_Supervisor\_Of$, $Major\_In$, etc.
- $\mathcal{T} = \{(e_{i_1}, r_{k_1}, e_{j_1}), (e_{i_2}, r_{k_2}, e_{j_2}), ...\}$: (e_i, r_k, e_j) are observed true facts about relationships between entities. E.g. $(Berlin, Is\_Capital\_Of, Germany)$, $(Prof\_X, Is\_Supervisor\_Of, Stud\_A)$, $(Stud\_A, Major\_In, Comp\_Science)$, etc.



(a) A multi-relational graph

(b) Intra-domain link prediction in a multi-relational graph

Figure 1: Figure 1a depicts a multi-relational graph about a university where the entities e_i are professors, students, staffs, research fields, and study objects, etc., and the predicates are types of relationships between these entities. Figure 1b shows an example of intra-domain link prediction in this graph. The entities of candidate triplets for prediction both belong to the same graph.

Figure 1a illustrates an example of multi-relational graph.

Link prediction is a problem of prime interest for graph-structured data. It is to predict if unconnected pairs of nodes have hidden links between them or not. In the context of multi-relational graph, the goal is to predict if an entity e_i has a relationship r_k with an entity e_j, i.e. to predict if an unobserved triplet (e_i, r_k, e_j) is true or not.

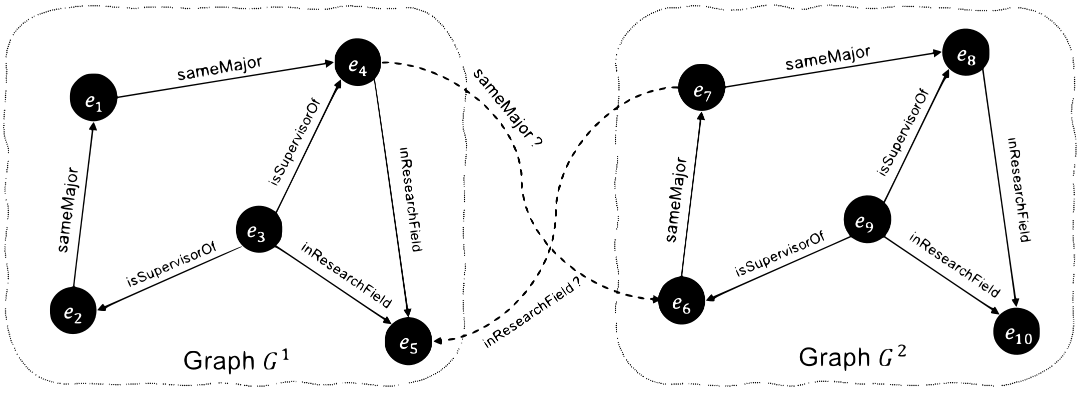


Figure 2: An example of inter-domain link prediction for two related graphs G^1 and G^2, which both are about relationships between personnel and study objects of different universities. The entities of candidate triplets for prediction lie on different graphs.

Conventional setting of the problem consider prediction for triplets (e_i, r_k, e_j) whose both entities e_i, e_j belongs to a same graph G (intra-domain link prediction), e.g. figure 1b. On the other hand, this study considers a different problem setting called inter-domain link prediction. It is motivated by the need to integrate graph data from different related sources. Let's say we have several graphs G^1, G^2, ..., G^l which all consider similar types of relationships between entities of the same kind. Naturally, we want to build a richer graph G by combining these related graphs. This requires predicting links between entities that lie not on the same graph but on different graphs G^i, e.g. figure 2. Examples of possible application scenarios are discussed in section 2.2.

2.2 Inter-domain link prediction problem

We give a formal definition of the inter-domain link prediction problem with the input and output as follows.

- **Input**: Two multi-relational graphs $G^1 = (\mathcal{E}^1, \mathcal{R}^1, \mathcal{T}^1)$ and $G^2 = (\mathcal{E}^2, \mathcal{R}^2, \mathcal{T}^2)$ with the entity sets $\mathcal{E}^t = \{e^t_1, ..., e^t_{n_t}\}$, the predicate sets $\mathcal{R}^t = \{r^t_1, ..., r^t_{m_t}\}$, and the sets of true facts $\mathcal{T}^t = \{(e^t_i, r^t_k, e^t_j)\}$, $t \in \{1, 2\}$ such that:
  - $\mathcal{R}^1$ and $\mathcal{R}^2$ are the same, i.e. $\mathcal{R}^1 \equiv \mathcal{R}^2 \equiv \mathcal{R} = \{r_1, ..., r_k\}$.
  - the entities of both graphs follow the same underlying distribution,

i.e. $\exists \boldsymbol{\pi}\ e_i^t \sim \boldsymbol{\pi}$.

  - $\mathcal{E}^1 \cap \mathcal{E}^2 = \mathcal{E}^c = \{e_1^c, ..., e_{n_c}^c\}$: The entity sets could be completely distinct ($n_c = 0$) or share some small amount of common entities ($n_c \ll n_1, n_2$). In the latter case, the identities of common entities are known.

- **Output**: For each inter-domain triplet (e_i^1, r_k, e_j^2) (or (e_i^2, r_k, e_j^1)), output a score of the likelihood that the triplet holds true.

In this problem setting, G^1, G^2 are not completely independent but rather closely related to each other. They share exactly the same set of predicates and their entity sets are distributionally similar. This relatedness is crucial for the feasibility of the problem.

Besides, we do not require the entity sets $\mathcal{E}^1$ and $\mathcal{E}^2$ to be strictly overlapped, they can share no common entities. However, a small amount of common entities are greatly beneficial to the proposed method as being seen in later experiments.

The following are some possible real scenarios where the distributional similarity is relatively satisfied and the above problem setting can be applied to integrate the graphs.

- G^i are graph data about personnel (professors, students, staffs) and objects (research fields, study objects) of different public universities in a country, e.g. Kyoto University, Tokyo University, Osaka University, etc. Since all public universities have similar structures with similar components, it can be assumed that the entity sets $\mathcal{E}^i$ follow the same distribution. Furthermore, overlapped entities could be research fields and study objects that are common between the universities.
- G^i are graph data collected on different populations of nearby areas, such as Kyoto, Osaka, Kobe, etc. Since the population live in the same sphere of similar cultural/socio-economic backgrounds (Kansai region), it is reasonable to assume that the entity sets $\mathcal{E}^i$ follow similar distributions to an extent. Overlapped entities could be people who live within the borders.
- Etc.

However, datasets for these examples are not available. Therefore, in the experiment, we create G^1 and G^2 by randomly sampling entities from a larger graph G.

To compute the prediction score for each inter-domain triplet, we follow the embedding approach. That is to learn suitable embedding representation $\mathbf{a}_\mathbf{i}^\mathbf{t}$ and $\mathbf{R}_k$ for each entity e_i^t and each predicate r_k, and learn a scoring function f to compute a score $f(\mathbf{a}_\mathbf{i}, \mathbf{R}_\mathbf{k}, \mathbf{a}_\mathbf{j})$ of a triplet (e_i, r_k, e_j). Due to the distributional similarity of $\mathcal{E}^1$ and $\mathcal{E}^2$, we are encouraged to learn $\mathbf{a}_i^t$ so that $\{\mathbf{a}_1^1, ..., \mathbf{a}_{n_1}^1\}$ and $\{\mathbf{a}_1^2, ..., \mathbf{a}_{n_2}^2\}$ also follow the same distribution.

3 Related Works

In recent years, the embedding-based approach has become popular in dealing with the link prediction task on a multi-relational knowledge graph (intra-domain). One of the pioneering works in this direction is TransE [7]. The model is inspired by the intuition from Word2Vec [8,9] that many predicates represent linear translations between entities in the latent embedding space, e.g. $\mathbf{a}_{\text{Japan}} - \mathbf{a}_{\text{Tokyo}} \approx \mathbf{a}_{\text{Germany}} - \mathbf{a}_{\text{Berlin}} \approx \mathbf{a}_{\text{is\_capital\_of}}$. Therefore, TransE tries to learn low-dimensional embedding vectors so that $\mathbf{a}_h + \mathbf{a}_r \approx \mathbf{a}_t$ for a true fact (e_h, r, e_t). TransE is suitable for 1-to-1 relationships only. Following translational models such as TransH, TransR, and TransD [10–12] are designed to deal with n-to-1, 1-to-n, and n-to-n relationships. Furthermore, tensor-based models such that RESCAL, DistMult, and SimplE [13–15] also gain huge interest. RESCAL converts a multi-relational graph data into a 3-D tensor whose first two modes indicate the entities and the third mode indicates the predicates. A low-rank decomposition technique is employed to compute embedding vectors of entities and embedding matrices of predicates. DistMult and SimplE further extend RESCAL by using diagonal matrices instead of full matrices to represent predicates and learning two embedding vectors dependently instead of one vector for each entity. Besides, neural network and complex vector-based models [16,17] are also introduced in the literature. Further details can be found in [18].

To the best of our knowledge, the proposed method is the first to consider the inter-domain link prediction problem between multi-relational graphs. Existing methods in the literature do not directly deal with the problem. The closest line of research focuses on entity alignment in multilingual knowledge graphs, which often aims to match words of the same meanings between different languages. The first work in this line of research is MTransE [19]. It employs TransE to independently embed different knowledge graphs and perform matching on the embedding spaces. Other methods like JAPE [20] and BootEA [21] further improve MTransE by exploiting additional attributes or description information and bootstrapping strategy. MRAEA [22] directly learns multilingual entity embeddings by attending over the entities' neighbors and their meta semantic information. Other methods [23, 24] apply Graph Neural Networks for learning alignment-oriented embeddings and achieve state-of-the-art results in many datasets. All these entity-matching methods implicitly assume most entities in one graph to have corresponding counterparts in the other graph, e.g. words in one lingual graph to have the same meaning words in the other lingual graph. Meanwhile, the proposed method only assumes the similarity between entity distributions.

Minimizing a dissimilarity criterion between distributions is a popular strategy for distribution matching and entity alignment problems. Cao et al. propose Distribution Matching Machines [6] that optimizes maximum mean discrepancy (MMD) between source and target domains for unsupervised domain adaptation tasks. The criterion is successfully applied in distribution matching and domain confusion tasks as well [25, 26]. Besides Wasserstein distance (WD), Gromov-Wasserstein distance (GWD) [27] also is a popular optimal transport metric. It measures the topological dissimilarity between distributions lying on different domains. GWD often requires much heavier computation than WD due to nested loops of Sinkhorn algorithm in current implementations [27]. Applying optimal transport into the graph matching problem, Xu et al. propose Gromov-Wasserstein Learning framework [1] for learning node embedding and node alignment simultaneously, and achieve state of the art in various graph matching datasets. Chen et al. [2] propose

Graph Optimal Transport framework that combines both WD and GWD for entity alignment. The framework is shown to be effective in many tasks such as image-text retrieval, visual question answering, text generation, and machine translation. Due to the computational complexity of GWD, each domain considered in [1,2] only contains less than several hundred entities. Phuc et al. [3] propose to apply WD to solve the intra-domain link prediction problem on two graphs simultaneously. In terms of technicality, the method is the most similar to the proposed method; however, it only focuses on the intra-domain link prediction problem on undirected homogeneous graphs and requires most of the nodes in one graph to have corresponding counterparts in the other graph.

4 Preliminary

This section briefly introduces the components that are employed in the proposed method.

4.1 RESCAL

RESCAL [13] formulates a multi-relational data as a three-way tensor $\mathbf{X} \in \mathbb{R}^{n\times n\times m}$, where n is the number of entities and m is the number of predicates. $\mathbf{X}_{i,j,k} = 1$ if the fact (e_i, r_k, e_j) exists and $\mathbf{X}_{i,j,k} = 0$ otherwise. In order to find proper latent embeddings for the entities and the predicates, RESCAL performs a rank-d factorization where each slice along the third mode $\mathcal{X}_k = \mathbf{X}_{\cdot,\cdot,k}$ is factorized as

$$\mathcal{X}_k \approx \mathbf{A}\mathbf{R}_k\mathbf{A}^\top, \text{ for } k = 1, ..., m.$$

Here, $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_n]^\top \in \mathbb{R}^{n\times d}$ contains the latent embedding vectors of the entities and $\mathbf{R}_k \in \mathbb{R}^{d\times d}$ is an asymmetric matrix that represents the interactions between entities in the k-th predicate.

Originally, it is proposed to learn $\mathbf{A}$ and $\mathbf{R}_k$ with the regularized squared loss function

$$\min_{\mathbf{A},\mathbf{R}_k} g(\mathbf{A}, \mathbf{R}_k) + \text{reg}(\mathbf{A}, \mathbf{R}_k),$$

where

$$g(\mathbf{A}, \mathbf{R}_k) = \frac{1}{2}\left(\sum_k \|\mathcal{X}_k - \mathbf{A}\mathbf{R}_k\mathbf{A}^\top\|_F^2\right)$$

and reg is the following regularization term

$$\text{reg}(\mathbf{A}, \mathbf{R}_k) = \frac{1}{2}\mu\left(\|\mathbf{A}\|_F^2 + \sum_k \|\mathbf{R}_k\|_F^2\right).$$

$\mu > 0$ is a hyperparameter.

It is later proposed by the authors of RESCAL to learn the embeddings with pairwise loss training [28], i.e. using the following margin-based ranking loss function

$$\min_{\mathbf{A}, \mathbf{R}_k} L(\mathbf{A}, \mathbf{R}_k) = \sum_{(e_i, r_k, e_j) \in \mathcal{D}^+} \sum_{(e_l, r_h, e_t) \in \mathcal{D}^-} \mathcal{L}(f_{ijk}, f_{lth}) + \text{reg}(\mathbf{A}, \mathbf{R}_k), \tag{1}$$

where $\mathcal{D}^+$ and $\mathcal{D}^-$ are the sets of all positive triplets (true facts) and all negative triplets (false facts), respectively. f_{ijk} denotes the score of (e_i, r_k, e_j), $f_{ijk} = f(\mathbf{a}_i, \mathbf{R}_k, \mathbf{a}_j) = \mathbf{a}_i^\top \mathbf{R}_k \mathbf{a}_j$ and $\mathcal{L}$ is the ranking function

$$\mathcal{L}(f^+, f^-) = \max(1 + f^- - f^+, 0).$$

The negative triplet set $\mathcal{D}^-$ is often generated by corrupting positive triplets, i.e. replacing one of the two entities in a positive triplet (e_i, r_k, e_j) with a randomly sampled entity.

The pairwise loss training aims to learn $\mathbf{A}$ and $\mathbf{R}_k$ so that the score f^+ of a positive triplet is higher than the score f^- of a negative triplet. Moreover, the margin-based ranking function is more flexible and easier to optimize with *stochastic gradient descent* (SGD) than the original squared loss function. In the proposed method, the pairwise loss training is adopted.

4.2 Optimal Transport

Optimal Transport (OT) provides a very powerful tool for comparing distributions. It was traditionally studied for the economic problem of transportation and allocations of resources. Given two piles of sand with the same weight and different shapes, OT aims to find the best way to move the sand from one pile to the other with the minimum total effort. In mathematical setting, that problem is cast as that of comparing two probability distributions. For a predefined cost of moving a unit mass in space, OT finds the best way to morph or *transport* the first distribution into the second distribution that has the minimum total transportation cost. If the predefined cost has nice properties, e.g. being a distance, the minimum total transportation cost defines a distance between probability distributions. Such a distance is often called the Wasserstein distance.

The exact computation of Wasserstein distance is often prohibitive, e.g. cubic time when using linear programming. To address this computational burden, a number of efficient methods for approximating the distance have been investigated based on primal and dual formulations of optimal transport.

4.2.1 Primal formulation

Let's consider two probability distributions $\boldsymbol{\pi}_1 = (\mathbf{p}^1, \mathbf{A}^1)$ and $\boldsymbol{\pi}_2 = (\mathbf{p}^2, \mathbf{A}^2)$, where $\mathbf{p}^1 \in \mathbb{R}_+^{n_1}$ and $\mathbf{p}^2 \in \mathbb{R}_+^{n_2}$ are probability vectors that satisfy ${\mathbf{p}^1}^\top \mathbb{1}_{n_1} = {\mathbf{p}^2}^\top \mathbb{1}_{n_2} = 1$ and $\mathbf{A}^1 = [\mathbf{a}_1^1, ..., \mathbf{a}_{n_1}^1]^\top \in \mathbb{R}^{n_1 \times d}$ and $\mathbf{A}^2 = [\mathbf{a}_1^2, ..., \mathbf{a}_{n_2}^2]^\top \in \mathbb{R}^{n_2 \times d}$ are the corresponding supports. Here, $\mathbb{1}_n$ indicates a n-dimensional vector of ones. Let a matrix $\mathbf{C} \in \mathbb{R}_+^{n_1 \times n_2}$ be a predefined transport cost matrix between the two distributions. For example, $\mathbf{C}$ can be defined as

$$C_{ij} = \|\mathbf{a}_i^1 - \mathbf{a}_j^2\|_2.$$

In its primal formulation, OT solves the following minimization problem.

$$\begin{aligned} \min_P \langle \mathbf{P}, \mathbf{C} \rangle &= \sum_{i,j} P_{ij} C_{ij} \\ \text{subject to} \quad & \mathbf{P}\mathbb{1}_{n_1} = \mathbf{p}^1 \\ & \mathbf{P}^\top \mathbb{1}_{n_2} = \mathbf{p}^2 \\ & \mathbf{P} \geq 0 \end{aligned} \tag{2}$$

A feasible matrix $\mathbf{P}$ that satisfies the problem's constraints is called a transport plan and the associated value $\langle \mathbf{P}, \mathbf{C} \rangle$ is called its transport cost. A transport plan $\mathbf{P}^*$ that gives the minimum transport cost, $\mathbf{P}^* = \arg\min_{\mathbf{P}} \langle \mathbf{P}, \mathbf{C} \rangle$, is called an optimal transport plan. The minimum cost $\langle \mathbf{P}^*, \mathbf{C} \rangle$ defines a distance between the two distributions $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$, which is often called the Wasserstein distance. The optimal transport plan $\mathbf{P}^*$ gives reasonable guidance of how to morph/transport one distribution into the other while the Wasserstein distance provides a measurement of how similar the two distributions are.

In the scope of multi-relational graphs, if one wants to compare two entity distributions, $\mathbf{p}^1$ and $\mathbf{p}^2$ could be predefined over the sets of entities, normally being set to be uniform and the entity embeddings could be seen as the supports $\mathbf{A}^1$ and $\mathbf{A}^2$.

A computation-efficient approach to approximate the optimal transport plan and Wasserstein distance has been proposed by Cuturi et al. [29]. Instead of the exact optimal transport $\mathbf{P}^*$, they compute an entropic-regularized optimal transport plan $\mathbf{P}^\lambda$ via minimizing a cost M as follows,

$$\mathbf{P}^\lambda = \arg\min_{\mathbf{P}} M(\mathbf{P}) = \langle \mathbf{P}, \mathbf{C} \rangle + \frac{1}{\lambda} \sum_{i,j} P_{ij} \log P_{ij}, \tag{3}$$

where $\lambda > 0$ is a hyperparameter controlling the effect of the negative entropy of matrix $\mathbf{P}$.

It is known that $\mathbf{P}^\lambda$ admits a unique representation of the following form

$$\mathbf{P}^\lambda = \mathbf{diag}(\mathbf{u})\mathbf{K}\mathbf{diag}(\mathbf{v}),$$

where **diag(u)** indicates a diagonal matrix whose diagonal elements are elements of $\mathbf{u}$. The matrix $\mathbf{K} = e^{-\lambda \mathbf{C}}$ is the element-wise exponential of $-\lambda$C. Vectors $\mathbf{u}$ and $\mathbf{v}$ can be computed via Sinkhorn iteration

$$(\mathbf{u}, \mathbf{v}) \leftarrow \left(\frac{\boldsymbol{\pi}_1}{\mathbf{K}\mathbf{v}}, \frac{\boldsymbol{\pi}_2}{\mathbf{K}^\top \mathbf{u}} \right).$$

The detailed computation of $\mathbf{P}^\lambda$ is summarized in algorithm 1.

With large enough λ, emperically when $\lambda > 50$, algorithm 1 converges quickly. The resulting $\mathbf{P}^\lambda$ and $\langle \mathbf{P}^\lambda, \mathbf{C} \rangle$ are highly accurate approximation of $\mathbf{P}^*$ and the Wasserstein distance.

Algorithm 1: Sinkhorn algorithm for computing $\mathbf{P}^{\lambda}$ [29]

```
Input: $\mathbf{C} \in \mathbb{R}_{+}^{n_1 \times n_2}$, $\mathbf{p}^1 \in \mathbb{R}_{+}^{n_1}$, $\mathbf{p}^2 \in \mathbb{R}_{+}^{n_2}$, $\lambda > 0$, thresh $> 0$
K = $e^{-\lambda \mathbf{C}}$; $\mathbf{v} = \mathbb{1}_{n_2}$
err = thresh
while err $\geq$ thresh do
    $\mathbf{u}_1 = \mathbf{u}$
    $\mathbf{u} = \frac{\mathbf{p}^1}{K\mathbf{v}}$
    $\mathbf{v} = \frac{\mathbf{p}^2}{K^T\mathbf{u}}$
    err = $\|\mathbf{u}_1 - \mathbf{u}\|_1$
end
$P^{\lambda} = \mathbf{diag}(\mathbf{u}) K \mathbf{diag}(\mathbf{v})$
Output: $P^{\lambda}$
```

4.2.2 Dual formulation

In its dual form, OT solves the following maximization problem which is the dual problem of (2).

$$\max_{\mathbf{f} \in \mathbb{R}^{n_1}, \mathbf{g} \in \mathbb{R}^{n_2}} \langle \mathbf{f}, \mathbf{p}^1 \rangle + \langle \mathbf{g}, \mathbf{p}^2 \rangle$$

$$\text{subject to} \quad f_i + g_j \leq C_{ij}$$

In the special case where $C_{ij} = \|\mathbf{a}_i^1 - \mathbf{a}_j^2\|_2$, the above problem can be rewrite as follows [30].

$$\begin{aligned} &\max_{f:\mathbb{R}^d \to \mathbb{R}} \mathbb{E}_{x \sim \boldsymbol{\pi}_1}[f(x)] - \mathbb{E}_{y \sim \boldsymbol{\pi}_2}[f(y)] \\ &\text{subject to} \quad \text{Lip}(f) \leq 1 \end{aligned} \tag{4}$$

Here, $\text{Lip}(f) = \sup \left\{ \frac{|f(x) - f(y)|}{\|x - y\|_2} : x, y \in \mathbb{R}^d \right\}$ is the Lipschitz constant of function f [1)].

Thanks to strong duality, the maximum of problem (4) is equal to the minimum of problem (2). Therefore, the Wasserstein distance can be computed by finding a function f that maximizes its value difference between two distribution $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ while satisfying the Lipschitz constant condition.

[1)] Optimizing over a function f is equivalent to optimizing over vectors $\mathbf{f}, \mathbf{g}$ in the maximization, because the maximum only depends on the values of f on the supports of $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$.

This problem is by no means easier than its primal counterpart. However, it could again be solved approximately.

Gulrajani et al. [31] proposed to approximate (4) with the following problem.

$$\max_{\Theta} \mathbb{E}_{x\sim\boldsymbol{\pi}_1}[f_\Theta(x)] - \mathbb{E}_{y\sim\boldsymbol{\pi}_2}[f_\Theta(y)] - \lambda\mathbb{E}_{\hat{x}\sim\boldsymbol{\pi}}(\|\nabla_{\hat{x}} f_\Theta(\hat{x})\|_2 - 1)^2 \tag{5}$$

Here, f_Θ is a neural network with parameter Θ. $\boldsymbol{\pi}$ defines sampling uniformly along straight lines between pairs of samples from $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$. The third term is the regularization for enforcing the Lipschitz constant condition. $\lambda > 0$ is a hyperparameter. This relaxed problem could be optimized by stochastic gradient descent with mini-batch sampling.

Besides (5), other approximations for the dual formulation have also been proposed in [32] for quadratic cost, $C_{ij} = \|\mathbf{a}_i^1 - \mathbf{a}_j^2\|_2^2$, using input convex neural networks and in [33] for a general cost C, to name a few.

Generally, approximating the Wasserstein distance with primal and dual formulation approaches both have their advantages and disadvantages. The dual formulation approach is highly scalable to very large data since it allows training with mini-batches; however, it gives little control and guarantee on how accurate the approximation is. On the other hand, the primal formulation approach allows highly accurate approximation, but it is not applicable in large data regimes due to quadratic computation time.

4.3 Maximum Mean Discrepancy

Maximum Mean Discrepancy (MMD) is another popular divergence for probability distributions, which is originally introduced as a non-parametric statistic to test if two distributions are different [4,5]. It is defined as the difference between mean function values on samples generated from the distributions. If MMD is large, the two distributions are likely to be distinct. On the other hand, if MMD is small, the two distributions can be seen to be similar.

Formally, let $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ be two distributions whose the supports are subsets of $\mathbb{R}^d$, and $\mathcal{F}$ be a class of functions $f : \mathbb{R}^d \to \mathbb{R}$. Usually, $\mathcal{F}$ is selected to be the unit ball in a universal RKHS $\mathcal{H}$. Then MMD is defined as

$$M(\mathcal{F}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{x \sim \boldsymbol{\pi}_1}[f(x)] - \mathbb{E}_{y \sim \boldsymbol{\pi}_2}[f(y)] \right).$$

From sample sets $\mathbf{A}^1 = \{\mathbf{a}_1^1, ..., \mathbf{a}_{n_1}^1\}$ and $\mathbf{A}^2 = \{\mathbf{a}_1^2, ..., \mathbf{a}_{n_2}^2\}$, $\mathbf{a}_i^t \in \mathbb{R}^d$, sampled from the two distributions, MMD can be unbiasedly approximated as follows [4,30].

$$\begin{aligned} M(\mathbf{A}^1, \mathbf{A}^2) =& \frac{1}{n_1(n_1-1)} \sum_{i,i'} k(\mathbf{a}_i^1, \mathbf{a}_{i'}^1) + \frac{1}{n_2(n_2-1)} \sum_{j,j'} k(\mathbf{a}_j^2, \mathbf{a}_{j'}^2) \\ &- \frac{2}{n_1 n_2} \sum_{i,j} k(\mathbf{a}_i^1, \mathbf{a}_j^2) \end{aligned} \tag{6}$$

Here, $k(\cdot, \cdot)$ is often chosen as the Gaussian kernel

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\sigma \|\mathbf{x} - \mathbf{y}\|_2^2), \sigma > 0,$$

even though other kernels such as the Laplacian, $\exp(-\sigma\|\mathbf{x} - \mathbf{y}\|_1)$, $\sigma > 0$, or the inverse multiquadratics, $(\sigma + \|\mathbf{x} - \mathbf{y}\|_2^2)^{-\frac{1}{2}}$, $\sigma > 0$, can also be used. When $\mathbf{A}^1$ and $\mathbf{A}^2$ are the embeddings of entities in two domains, MMD represents a dissimilarity between the domains' entity distributions.

5 Proposed Method

5.1 Proposed objective function

The proposed method's objective function consists of two components. The first component is for learning embedding representations of the entities and the predicates of each multi-relational graph, which is based on an existing tensor-factorization method. RESCAL [13] is specifically chosen in the proposed method due to its simplicity and generally competitive performance. The second component is a regularization term for enforcing the entity embedding distributions of the two graphs to become similar.

For each graph G^t, lets denote the entity embeddings as $\mathbf{A}^t = [\mathbf{a}_1^t, ..., \mathbf{a}_{n_t}^t]^\top \in \mathbb{R}^{n_t \times d}$, where d is the embedding dimension. If the entity sets $\mathcal{E}^1$ and $\mathcal{E}^2$ overlap, the embeddings of common entities are set to be identical in both domains, i.e. $\mathbf{A}^t = [\mathbf{A}'^t, \mathbf{A}_c]^\top$ where $\mathbf{A}_c \in \mathbb{R}^{d \times n_c}$ is the embeddings of common entities. The embedding of predicate $r_k \in \mathcal{R}$ is denoted as $\mathbf{R}_k \in \mathbb{R}^{d \times d}$

for $k \in \{1, ..., m\}$. The objective function of the proposed method is given as

$$F(\mathbf{A}^1, \mathbf{A}^2, \mathbf{R}_k, [\mathbf{P}]) = L(\mathbf{A}^1, \mathbf{R}_k) + L(\mathbf{A}^2, \mathbf{R}_k) + \alpha M(\mathbf{A}^1, \mathbf{A}^2, [\mathbf{P}]). \qquad (7)$$

In (7), the first two terms $L(\mathbf{A}^t, \mathbf{R}_k)$ are the loss functions of RESCAL and are defined as in (1). The third term $M(\mathbf{A}^1, \mathbf{A}^2, [\mathbf{P}])$ is the Wasserstein distance (WD) or the MMD discrepancy between the entity distributions of the two graphs. In the case of WD regularizer, the primal formulation with entropic regularization is used to approximate the distrance, i.e. $M = M(\mathbf{A}^1, \mathbf{A}^2, \mathbf{P})$ as defined in (3) with $\mathbf{P} \in \mathbb{R}_+^{n_1 \times n_2}$. In the case of MMD regularizer, $M = M(\mathbf{A}^1, \mathbf{A}^2)$ is defined as in (6).

The objective function $F(\mathbf{A}^1, \mathbf{A}^2, \mathbf{R}_k)$ (MMD regularizer) is directly optimized with SGD. On the other hand, $F(\mathbf{A}^1, \mathbf{A}^2, \mathbf{R}_k, \mathbf{P})$ (WD regularizer) is minimized iteratively. In each epoch, the transport plan $\mathbf{P}$ is fixed, and the embedding vectors $\mathbf{A}^1$, $\mathbf{A}^2$, and $\mathbf{R}_k$ are updated with SGD. At the end of each epoch, $\mathbf{A}^1$, $\mathbf{A}^2$, and $\mathbf{R}_k$ are fixed and the plan $\mathbf{P}$ is sequentially updated via algorithm 1.

Via $L(\mathbf{A}^t, \mathbf{R}_k)$, the underlying distribution governing $\mathcal{E}^t$ is learned and characterized into $\mathbf{A}^t$. In the latent space $\mathbb{R}^d$, embedding vectors $\mathbf{a}_i^t$ of entities with similar roles (entities who engage in many same kinds of relationships, e.g. professors in the same department engage in many similar interactions/relationships towards students of the same department) lie close together. Since $\mathcal{E}^1$ and $\mathcal{E}^2$ follow the same distribution, the learned embedding distributions $\{\mathbf{a}_1^1, ..., \mathbf{a}_{n_1}^1\}$ and $\{\mathbf{a}_1^2, ..., \mathbf{a}_{n_2}^2\}$ are expected to be in similar layouts/shapes. However, they do not necessarily stay in the same absolute location in the latent space due to randomness in optimization and initialization of $L(\mathbf{A}^t, \mathbf{R}_k)$ (e.g. figures 4a).

Minimizing the dissimilarity $M(\mathbf{A}^1, \mathbf{A}^2, [\mathbf{P}])$ helps to enhance the similarity both in shapes and absolute positions of the embedding distributions. In theory, it could be possible to "pull" the distributions closer via minimizing M even if they are initially disjoint and away in space (a possible scenario if the node sets are distinct), but the optimization will likely be unstable and difficult in practice. When the two node sets share even some small amount of entities, embedding vectors of the common entities act as anchors to hold the embedding distributions in close proximity right from the beginning of optimization. This makes the optimization process become easier. We will see in the later experiments that while it does work in some scenarios of distinct node sets, the proposed method favors overlapping scenarios more.

Through optimizing both $L(\mathbf{A}^t, \mathbf{R}_k)$ and $M(\mathbf{A}^1, \mathbf{A}^2, [\mathbf{P}])$, entities with similar roles in G^1 and G^2 are expected to lie close together in the latent embedding space. This is the intuition of the proposed method's inter-domain link prediction. If $e_i^1 \in \mathcal{E}^1$ and $e_i^2 \in \mathcal{E}^2$ have similar embeddings $\mathbf{a}_i^1$ and $\mathbf{a}_i^2$, the inter-domain fact (e_i^1, r_k, e_j^2) is likely to exist if the intra-domain fact (e_i^2, r_k, e_j^2) exists thanks to their similar scores $\mathbf{a}_i^{1\top}\mathbf{R}_k\mathbf{a}_j^2 \approx \mathbf{a}_i^{2\top}\mathbf{R}_k\mathbf{a}_j^2$.

5.2 Discussion of possible variants

Besides using MMD and the primal formulation of optimal transport, it is also possible to use the dual formulation of optimal transport to compute the term $M(\mathbf{A}_1, \mathbf{A}_2)$ in (7). In this case, the objective F has the following form

$$F(\mathbf{A}^1, \mathbf{A}^2, \mathbf{R}_k) = L(\mathbf{A}^1, \mathbf{R}_k) + L(\mathbf{A}^2, \mathbf{R}_k) + \alpha(\mathbb{E}[f_\Theta(\mathbf{a}_i^1)] - \mathbb{E}[f_\Theta(\mathbf{a}_j^2)]),$$

where f_Θ is the neural network that maximizes (5). The optimization of F then involves an iterative min-max procedure, i.e. alternate between approximately maximizing (5) to find f_Θ and minimize F with respect to the embeddings.

As mentioned in section 4.2.2, dual formulation of optimal transport does not guarantee accurate approximation of Wasserstein distance. The error will likely increase further if one only solves (5) approximately in the min-max optimization. Therefore, using the dual approximation of Wasserstein distance might pose a challenge for F to learn $\mathbf{A}^1$ and $\mathbf{A}^2$ with similar distributions. However, on the other hand, it allows scalability to graphs with a larger number of entities than the primal approximation.

Other variants of (7) could come from different choices of embedding methods besides RESCAL. Different methods design different scoring functions f for their loss objectives L. DisMult [14] and SimplE [15] use similar functions as RESCAL, $f_{\text{DisMult}} = \mathbf{a}_i^\top \mathbf{diag}(\mathbf{r}_k)\mathbf{a}_j$ and $f_{\text{SimplE}} = \frac{1}{2}(\mathbf{a}_i^\top \mathbf{diag}(\mathbf{r}_k)\mathbf{a}_j + \mathbf{a'}_i^\top \mathbf{diag}(\mathbf{r'}_k)\mathbf{a'}_j)$. Translational models like TransE [7] choose a transitional difference of the embedding vectors as its function, $f_{\text{TransE}} = -\|\mathbf{a}_i + \mathbf{r}_k - \mathbf{a}_j\|_{1,2}$. The neural tensor network models like NTN [16] generalize RESCAL's approach by combining traditional MLPs and bilinear operators in its scoring function, $f_{\text{NTN}} = \mathbf{r}_{1k}^\top \tanh(\mathbf{a}_i^\top \mathbf{R}_{1k}\mathbf{a}_j + \mathbf{R}_{2k}\mathbf{a}_i + \mathbf{R}_{3k}\mathbf{a}_j + \mathbf{r}_{2k})$. More complicated scoring functions are also employed in other embedding methods [18]. Nevertheless, as long as an embedding method could characterize entity distributions with its learned embeddings, it is a feasible choice for the proposed method.

Further investigation on variants of the proposed objective (7) with other embedding methods and dual approximation of Wasserstein distance might be an interesting research direction, which we leave for future works.

6 Experiments

6.1 Datasets

The datasets used in the experiments are created from four popular knowledge graph datasets, namely FB15k-237 [34], WN18RR [35], DBbook2014, and ML1M [36]. The FB15k-237 dataset contains $272k$ facts about general knowledge. It has $14k$ entities and 237 predicates. The WN18RR dataset consists of $86k$ facts about 11 lexical relations between $40k$ word senses. The other two datasets represent interactions among users and items in e-commerce. The ML1M (MovieLens-1M) dataset composes of $434k$ facts with $14k$ users/items and 20 relations, while the DBbook2014 has $334k$ facts with $13k$ users/items and 13 relations. To create G^1 and G^2 for each dataset, two smaller sub-graphs of around $2k$ to $3k$ entities are randomly sampled from the original graph. The two graphs are controlled to share some amounts of common entities. Different levels of entity overlapping are investigated, from 0% (non-overlapping setting) to around $1.5\%, 3\%$, and 5% (overlapping setting). Moreover, different predicates are removed so that G^1 and G^2 share the same predicate set, i.e. $\mathcal{R}^1 \equiv \mathcal{R}^2 \equiv \mathcal{R}$.

Intra-domain triplets (e_i, r_k, e_j) whose both entities e_i, e_j belong to the same graph are used for training. Inter-domain triplets (e_i, r_k, e_j) whose entities e_i, e_j belong to different graphs are used for validating and testing inter-domain performance. The validation and test ratio is $20 : 80$. Even though the goal is to evaluate a model's ability to perform inter-domain link prediction, both inter-domain and intra-domain link prediction performances are evaluated. This is because the proposed method should improve inter-domain link prediction while does not harm intra-domain link prediction. Therefore, 5% of intra-domain triplets are further spared from the training data for monitoring intra-domain performance.

The details for the case of 3% overlapping are shown in Table 1. In other cases, the datasets share similar statistics.

Table 1: Details of the datasets in the case of 3% overlapping. The other cases share similar statistics.

Datasets	#Ent G1	#Ent G2	#Rel	#Train	#Inter Valid	#Intra Test	#Inter Test
FB15k-237	2675	2677	179	24.3k	4.3k	1.3k	17.7k
WN18RR	2804	2720	10	5.1k	105	148	1.1k
DBbook2014	2932	2893	11	34.6k	6.5k	1.8k	26.8k
ML1M	2764	2726	18	39.3k	6.5k	2k	27k

6.2 Evaluation methods and Baselines

In the experiments, Hit@10 score and ROC-AUC score are used for quantifying both inter-domain and intra-domain performances.

6.2.1 Evaluation with Hit@10

The Hit@10 score is computed by ranking true entities based on their scores. For each true triplet (e_i, r_k, e_j) in the test sets, one entity e_i (or e_j) is hidden to create an unfinished triplet $(\cdot, r_k, e_j)$ (or $(e_i, r_k, \cdot)$). All entities e_{cand} are used as candidates for completing the unfinished triplet and the scores of $(e_{\text{cand}}, r_k, e_j)$ (or $(e_i, r_k, e_{\text{cand}})$) are computed. Note that the candidates e_{cand} are taken from the same entity set as e_i (or e_j), i.e. if e_i (or e_j) $\in \mathcal{E}^t$ then entities

e_{cand} are taken from $\mathcal{E}^t$. The ranking of e_i (or e_j) is computed according to the scores. The higher "true" entities are ranked the better a model is at predicting hidden true triplets. Hit@10 score is used for quantifying the link prediction performance and is calculated as the percentage of "true" entities being ranked inside the top 10.

6.2.2 Evaluation with ROC-AUC

In order to compute the ROC-AUC score, triplets in the test set are treated as positive samples. An equal number of triplets are uniformly sampled from the entity sets and the predicate set to create negative samples. Due to the sparsity of each graph, it is safe to consider the sampled triplets as negative. During the sampling process, both sampled entities are controlled to belong to the same graph in the intra-domain case and belong to different graphs in the inter-domain case.

6.2.3 Evaluated Models

In the experiments, RESCAL is used as the baseline method. The proposed method with Wasserstein regularization based on the primal formulation is denoted as WD while the one with MMD regularization is denoted as MMD.

6.3 Implementation details

6.3.1 Negative sampling

Only intra-domain negative triplets are used in order to train the pairwise ranking loss (1) with SGD, i.e. negative triplet set $\mathcal{D}^-$ only contains negative triplets (e_l, r_h, e_t) whose both entities belong to the same graph.

6.3.2 Warmstarting

Completely learning from scratch might be difficult since the regularizer M can add noise at the early state. Instead, it is beneficial to warmstart the proposed method's embeddings with embeddings roughly learned by RESCAL. Specifically, we run RESCAL for 100 epochs to learn initial embeddings. After that, to maintain the fairness of equal training time, both the proposed method and RESCAL are warmstarted with the roughly learned embeddings.

6.3.3 Hyperparameters.

In the implementation, the latent embedding dimension is set to equal 100. All experiments are run for 300 epochs. Early stopping is employed with a patience budget of 50 epochs. Other hyperparameters, namely α, learning rate, and batch size, are tuned on the inter-domain validation set using Optuna [37]. During the tuning process, α is sampled to be between 0.5 and 10.0, while the learning rate and batch size are chosen from $\{0.01, 0.005, 0.001, 0.0005\}$ and $\{100, 300, 500, 700\}$, respectively. The hyperparameters of RESCAL is tuned similarly with fixed $\alpha = 0.0$. The kernel used in MMD is set to be a mixture of Gaussian kernels with the bandwidth list of $[0.25, 0.5, 1., 2., 4.] * c$ where c is the mean Euclidean distance between the entities. All results are averaged over 10 random runs[1)].

6.4 Experimental results

The experimental results are shown in Tables 2, 3, 4, and 5. Note that a random predictor has a Hit@10 score of less than 0.004 and a ROC-AUC score of around 0.5.

6.4.1 Inter-domain results

As being demonstrated in tables 2 and 3, the proposed method with WD regularizer works well with the FB15k-237 dataset, which outperforms RESCAL in all settings. Especially in the overlapping cases where few entities are shared between the graphs, both Hit@10 and ROC-AUC scores are improved significantly. The WD regularizer also demonstrates its usefulness with the DBbook2014 and ML1M datasets. The Hit@10 scores are boosted up in most cases of overlapping settings, while the ROC-AUC scores are consistently enhanced over that of RESCAL. Most of the time, the improvements are considerable. However, for the case of the ML1M dataset with 3% overlapping entities, the WD regularizer causes the Hit@10 score to deteriorate, from 0.230 to 0.213. On the other hand, the MMD regularizer seems not to be beneficial for the task. Unexpectedly, the regularizer introduces noise and reduces

[1)] The code is available at `https://github.com/phucdoitoan/inter-domain_lp`

Table 2: **Inter-domain Hit@**10 scores. Italic numbers indicate better results while bold numbers and bold numbers with asterisk $*$ indicate better results at significance level $p = 0.1$ and $p = 0.05$, respectively. The proposed method with WD regularizer achieves better scores in many settings.

Overlapping	Model	FB15k-237	WN18RR	DBbook2014	ML1M
0%	RESCAL	0.110 ±0.038	0.027 ±0.003	0.087 ±0.058	0.062 ±0.074
	MMD	0.111 ±0.038	0.031 ±0.004	0.085 ±0.057	0.063 ±0.072
	WD	**0.145 ±0.063**	0.024 ±0.004	0.084 ±0.070	0.061 ±0.067
1.5%	RESCAL	0.251 ±0.031	0.025 ±0.002	0.107 ±0.035	0.210 ±0.034
	MMD	0.237 ±0.043	0.026 ±0.003	0.109 ±0.037	0.180 ±0.067
	WD	**0.291 ±0.031***	0.024 ±0.002	*0.128 ±0.059*	**0.240 ±0.031***
3%	RESCAL	0.302 ±0.020	0.028 ±0.004	0.266 ±0.056	**0.230 ±0.003***
	MMD	0.292 ±0.020	0.026 ±0.004	0.227 ±0.081	0.228 ±0.002
	WD	**0.328 ±0.011***	0.025 ±0.004	**0.318 ±0.066***	0.213 ±0.006
5%	RESCAL	0.339 ±0.007	0.027 ±0.005	0.389 ±0.032	0.237 ±0.011
	MMD	0.334 ±0.006	0.026 ±0.004	0.388 ±0.027	0.236 ±0.010
	WD	**0.361 ±0.010***	0.031 ±0.004	0.389 ±0.051	**0.256 ±0.006***

the accuracy of inter-domain link prediction. In the case of the WN18RR dataset, both RESCAL and the proposed method fail to perform, in which all Hit@10 and ROC-AUC scores are close to random. This might be due to the extreme sparsity of the dataset, whose amount of observed triplets is only about one-fifth of that of the other datasets.

In all the four datasets, sharing some common entities, even with a small number, is helpful and important for predicting inter-domain links. These common entities act as anchors between the graphs, which guide the regularizer to learn similar embedding distributions. Without common entities, the learning process becomes more challenging and often results in uncertain predictors as being shown in the 0% overlapping cases. The overlapping scenarios are reasonable because, in practice, two related graphs often share some amounts of common entities, and identifying these common entities in a small quantity is not expensive.

Table 3: **Inter-domain ROC-AUC** scores. Italic numbers indicate better results while bold numbers and bold numbers with asterisk $*$ indicate better results at significance level $p = 0.1$ and $p = 0.05$, respectively. The proposed method with WD regularizer achieves better scores in many settings.

Overlapping	Model	FB15k-237	WN18RR	DBbook2014	ML1M
0%	RESCAL	0.504 ±0.092	0.504 ±0.009	0.483 ±0.097	0.464 ±0.173
	MMD	0.507 ±0.093	0.500 ±0.010	0.485 ±0.095	0.480 ±0.172
	WD	*0.548 ±0.118*	0.505 ±0.009	0.488 ±0.099	0.495 ±0.179
1.5%	RESCAL	0.793 ±0.044	0.512 ±0.009	0.640 ±0.066	0.805 ±0.027
	MMD	0.770 ±0.063	0.507 ±0.009	0.632 ±0.063	0.754 ±0.087
	WD	**0.837 ±0.033***	0.510 ±0.007	*0.671 ±0.087*	**0.842 ±0.017***
3%	RESCAL	0.825 ±0.022	0.503 ±0.009	0.762 ±0.032	0.832 ±0.006
	MMD	0.813 ±0.030	0.498 ±0.011	0.714 ±0.060	0.831 ±0.007
	WD	**0.850 ±0.013***	0.502 ±0.013	**0.809 ±0.030***	0.840 ±0.008
5%	RESCAL	0.870 ±0.008	0.498 ±0.021	0.824 ±0.012	0.845 ±0.007
	MMD	0.875 ±0.007	0.498 ±0.012	0.823 ±0.015	0.845 ±0.006
	WD	**0.902 ±0.010***	0.498 ±0.013	**0.835 ±0.020**	**0.867 ±0.003***

6.4.2 Intra-domain results

Even though the main goal is to predict inter-domain links, it is preferable that the regularizers do not harm performance on intra-domain link prediction when fusing the two domains' entity distributions. As being demonstrated in table 5, the proposed method is able to maintain similar or better intra-domain ROC-AUC scores compared to RESCAL. However, it sometimes requires trade-offs in terms of the Hit@10 score, which is shown in table 4. Specifically, the WD regularizer worsens the intra-domain Hit@10 scores compared to RESCAL in FB15k-237 with 5% overlapping and ML1M with 1.5% overlapping settings despite helping improve the inter-domain counterparts. It also hurts the intra-domain Hit@10 score in ML1M with 3% overlapping setting.

Table 4: **Intra-domain Hit@**10 scores. Bold numbers with asterisk $*$ indicate better results at significance level $p = 0.05$. Generally, the proposed method with WD regularizer preserves the intra-domain Hit@10 scores despite requiring trade-offs in some cases.

Overlapping	Model	FB15k-237	WN18RR	DBbook2014	ML1M
0%	RESCAL	0.451 ±0.031	0.418 ±0.031	0.468 ±0.011	0.302 ±0.076
	MMD	0.461 ±0.029	0.342 ±0.086	0.449 ±0.012	0.307 ±0.070
	WD	0.469 ±0.019	0.421 ±0.032	0.472 ±0.014	0.332 ±0.027
1.5%	RESCAL	0.433 ±0.008	0.390 ±0.040	0.296 ±0.039	**0.425 ±0.006***
	MMD	0.438 ±0.008	0.330 ±0.067	0.328 ±0.027	0.423 ±0.036
	WD	0.427 ±0.009	0.408 ±0.035	0.291 ±0.038	0.412 ±0.008
3%	RESCAL	0.433 ±0.009	0.476 ±0.074	0.413 ±0.008	**0.447 ±0.006***
	MMD	0.447 ±0.011	0.485 ±0.074	0.411 ±0.017	0.444 ±0.008
	WD	0.439 ±0.009	**0.620 ±0.026***	0.412 ±0.009	0.413 ±0.021
5%	RESCAL	**0.433 ±0.009***	0.455 ±0.038	0.418 ±0.010	0.408 ±0.005
	MMD	0.421 ±0.009	0.416 ±0.058	0.420 ±0.014	0407 ±0.004
	WD	0.413 ±0.007	0.479 ±0.076	0.412 ±0.022	0.401 ±0.005

6.4.3 Summary

The proposed method with WD regularizer significantly improves the performance of inter-domain link prediction over the baseline method while being able to preserve the intra-domain performance in the FB15k-237 and DBbook2014 datasets. In the ML1M dataset, it benefits the inter-domain performance at the risk of decreasing intra-domain Hit@10 scores. Unexpectedly, the MMD regularizer does not work well and empirically causes deterioration of the inter-domain performance. These negative results might be due to local optimal arising when minimizing MMD with a finite number of samples, as recently studied in [38]. Further detailed analysis would be necessary before one can firmly judge the performance of the MMD regularizer. We leave this matter for future works. It is also worth mentioning that, in the experiment setting, the sampling of G^1 and G^2 is repeated independently for each overlapping level. Therefore, it is not necessary for the link prediction scores to

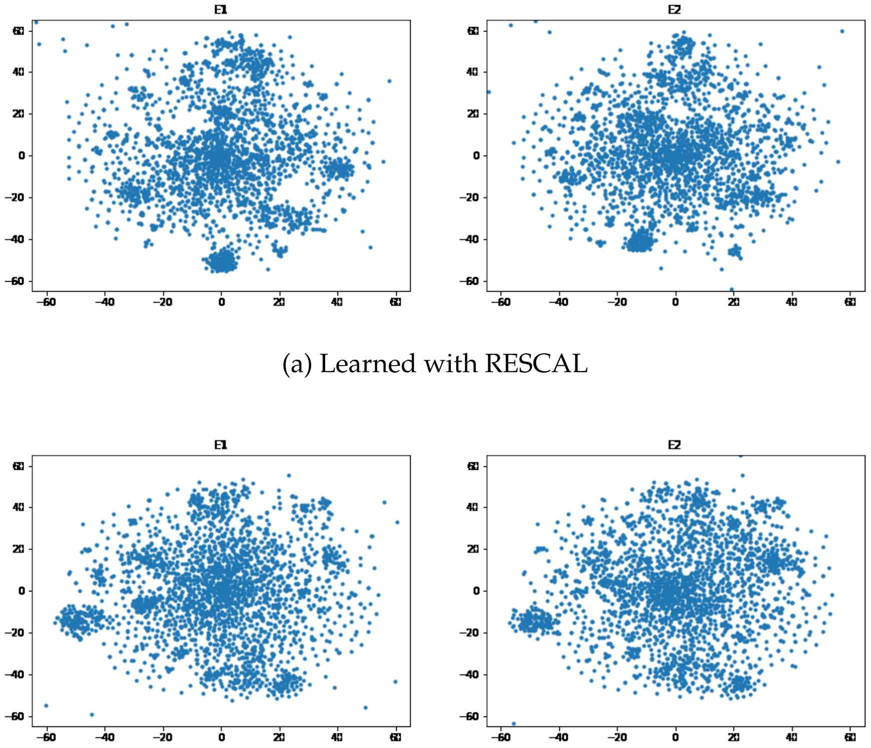
Table 5: **Intra-domain ROC-AUC** scores. Bold numbers with asterisk $*$ indicate better results at significance level $p = 0.05$. The propose method maintains similar or better intra-domain ROC-AUC scores compared to RESCAL.

Overlapping	Model	FB15k-237	WN18RR	DBbook2014	ML1M
0%	RESCAL	0.925 ±0.018	0.819 ±0.018	0.915 ±0.004	0.897 ±0.022
	MMD	0.924 ±0.018	0.818 ±0.019	0.915 ±0.005	0.897 ±0.035
	WD	0.928 ±0.006	0.811 ±0.017	0.918 ±0.005	**0.932 ±0.004***
1.5%	RESCAL	0.929 ±0.003	0.814 ±0.018	0.871 ±0.032	0.950 ±0.003
	MMD	0.931 ±0.003	0.807 ±0.029	0.892 ±0.009	0.954 ±0.003
	WD	0.932 ±0.006	0.818 ±0.020	0.868 ±0.040	0.954 ±0.002
3%	RESCAL	0.922 ±0.006	0.870 ±0.018	0.885 ±0.008	0.946 ±0.005
	MMD	0.926 ±0.005	0.861 ±0.011	0.877 ±0.026	0.948 ±0.003
	WD	0.921 ±0.007	0.860 ±0.018	0.890 ±0.005	0.949 ±0.003
5%	RESCAL	0.927 ±0.007	0.869 ±0.007	0.878 ±0.008	0.949 ±0.003
	MMD	0.935 ±0.005	0.835 ±0.050	0.879 ±0.008	0.952 ±0.003
	WD	**0.937 ±0.004***	0.860 ±0.020	**0.885 ±0.009***	0.953 ±0.003

monotonically increase when the overlapping level increases.

6.4.4 Embedding visualization

Figures 3, 4, 5 and 6 visualize the entity embeddings learned by RESCAL and the WD regularizer in the case of 3% overlapping. As being seen in figures 3 and 4, WD can learn more identical embedding distributions than RESCAL in the case of the FB15k-237 and DBbook2014 datasets. Especially, in the DBbook2014 dataset, RESCAL can only learn similar shape distributions, but the regularizer can learn distributions with both similar shape and close absolute position. However, as being shown in figures 5 and 6, in the WN18RR and ML1M datasets, the WD regularizer seems to only add noise when learning the embeddings, which results in no improvement or even degradation of both intra-domain and inter-domain Hit@10 scores.



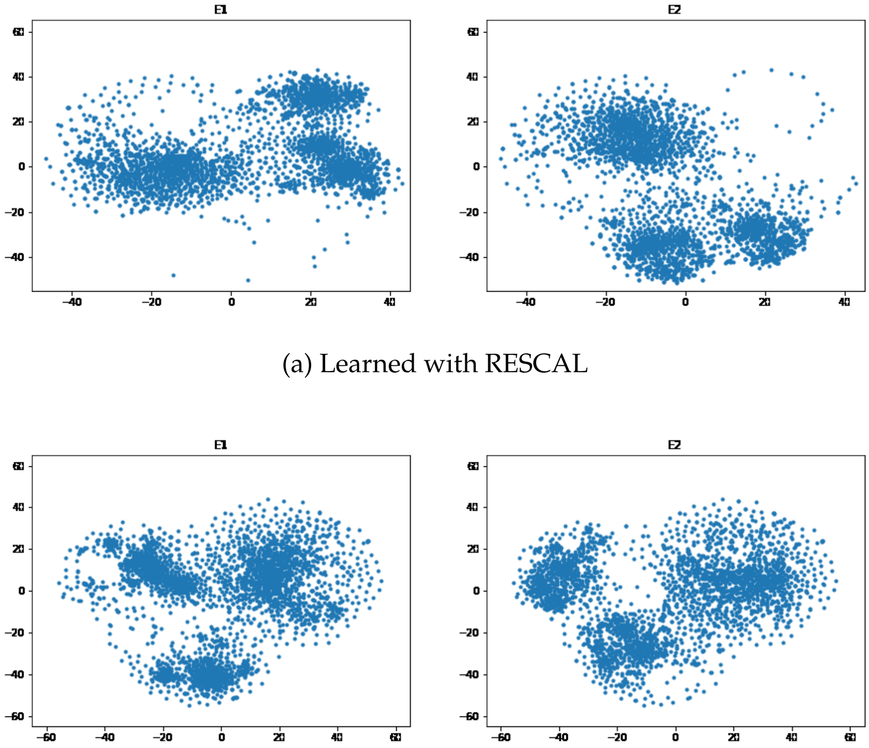
(a) Learned with RESCAL

(b) Learned with WD regularizer

Figure 3: Embedding visualization of FB15k-237 datasets with 3% overlapping. Figure 3a demonstrates entity embeddings learned by RESCAL while figure 3b depicts entity embeddings learned by the proposed method with WD regularizer. The proposed method learns more identical embedding distributions across both domains.

7 Conclusion and Future Work

Inter-domain link prediction is an important task for constructing large multi-relational graphs from smaller related ones. However, existing methods in the literature do not directly address this problem. In this paper, we propose a new approach for the problem via jointly minimizing a divergence between entity distributions during the embedding learning process. Two regularizers have been investigated, in which the regularizer based on primal approximation of Wasserstein distance shows promising results and improves inter-domain link prediction performance considerably. For future works, we

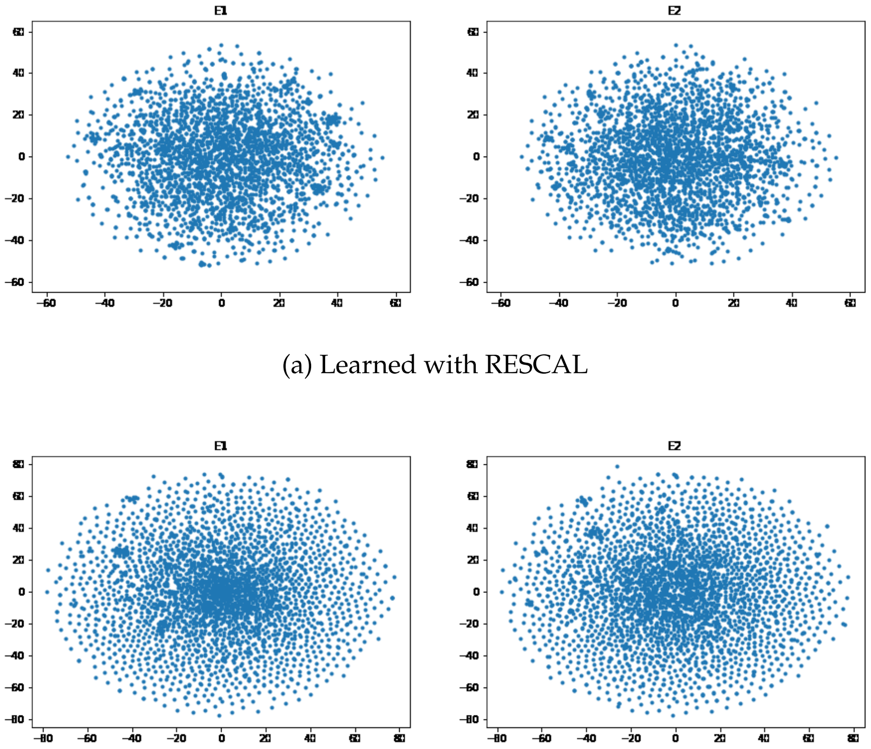


(a) Learned with RESCAL

(b) Learned with WD regularizer

Figure 4: Embedding visualization of DBbook2014 datasets with 3% overlapping. Figure 4a demonstrates entity embeddings learned by RESCAL while figure 4b depicts entity embeddings learned by the proposed method with WD regularizer. The proposed method learns more identical embedding distributions across both domains.

would like to verify the proposed method's effectiveness using more baseline embedding methods besides RESCAL. Further analysis on the performance of the MMD-based regularizer will also be conducted. To improve the method's scalability to larger graphs, we plan to study the use of dual approximation of Wasserstein distance as discussed in section 5.2. Last but not least, the proposed method currently assumes that both domains share the same underlying entity distribution. This assumption is violated when the domains' entity distributions are not completely identical but partially different, which is likely to happen in practice. One possible direction for further research is to adopt unbalanced optimal transport as the regularizer, which flexibly allows



(a) Learned with RESCAL

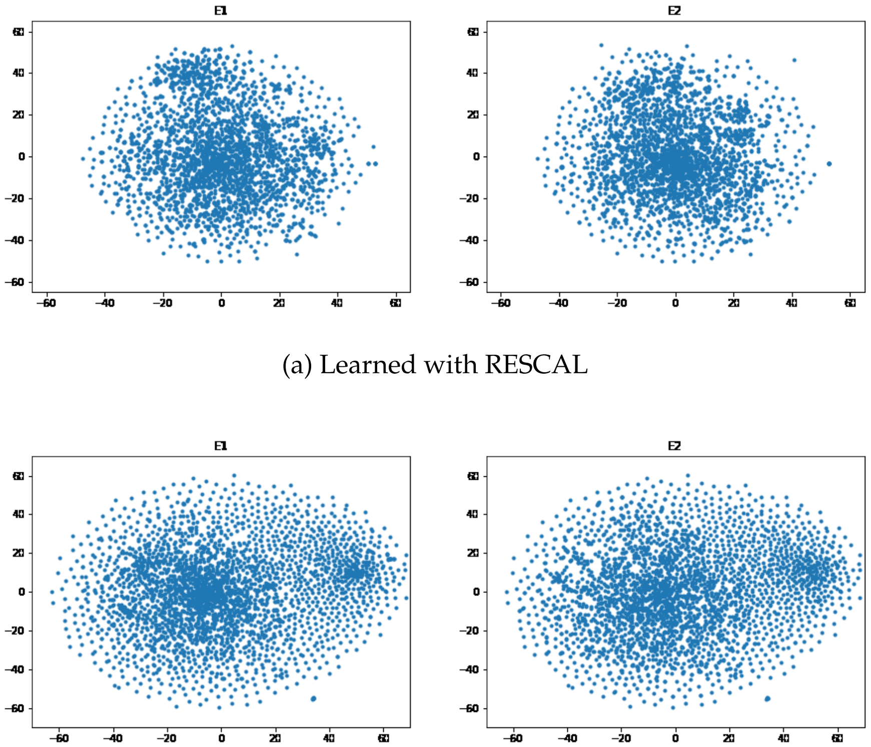
(b) Learned with WD regularizer

Figure 5: Embedding visualization of WN18RR datasets with 3% overlapping. Figure 5a demonstrates entity embeddings learned by RESCAL while figure 5b depicts entity embeddings learned by the proposed method with WD regularizer. RESCAL is able to learn similar embedding distributions between the two domains while the proposed method seems to add more noise.

mass destruction and mass creation to deal with distributions' differences.

Acknowledgments

I wish to express my sincere thanks to my supervisor, Professor Hisashi Kashima. Without his thorough instructions, immense support, and patience, I would have not been able to complete this research. I also would like to thank the members of our laboratory for their helpful comments and constructive feedbacks.



(a) Learned with RESCAL

(b) Learned with WD regularizer

Figure 6: Embedding visualization of ML1M datasets with 3% overlapping. Figure 6a demonstrates entity embeddings learned by RESCAL while figure 6b depicts entity embeddings learned by the proposed method with WD regularizer. RESCAL is able to learn similar embedding distributions between the two domains while the proposed method seems to add more noise.

References

[1] Xu, H., Luo, D., Zha, H. and Carin, L.: Gromov-Wasserstein Learning for Graph Matching and Node Embedding, *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6932–6941 (2019).

[2] Chen, L., Gan, Z., Cheng, Y., Li, L., Carin, L. and Liu, J.: Graph Optimal Transport for Cross-Domain Alignment, *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 1542–1553 (2020).

[3] Phuc, L. H., Takeuchi, K., Yamada, M. and Kashima, H.: Simultaneous Link Prediction on Unaligned Networks Using Graph Embedding and

Optimal Transport, *Proceedings of the Seventh IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 245–254 (2020).

[4] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. and Smola, A. J.: A Kernel Method for the Two-Sample-Problem, *Advances in Neural Information Processing Systems 19*, pp. 513–520 (2006).

[5] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. and Smola, A. J.: A Kernel Approach to Comparing Distributions, *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1637–1641 (2007).

[6] Cao, Y., Long, M. and Wang, J.: Unsupervised Domain Adaptation With Distribution Matching Machines, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2795–2802 (2018).

[7] Bordes, A., Usunier, N., García-Durán, A., Weston, J. and Yakhnenko, O.: Translating Embeddings for Modeling Multi-relational Data, *Advances in Neural Information Processing Systems 26*, pp. 2787–2795 (2013).

[8] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *Proceedings of the First International Conference on Learning Representations (ICLR)* (2013).

[9] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *Advances in Neural Information Processing Systems 26*, pp. 3111–3119 (2013).

[10] Wang, Z., Zhang, J., Feng, J. and Chen, Z.: Knowledge Graph Embedding by Translating on Hyperplanes, *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1112–1119 (2014).

[11] Lin, Y., Liu, Z., Sun, M., Liu, Y. and Zhu, X.: Learning Entity and Relation Embeddings for Knowledge Graph Completion, *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2181–2187 (2015).

[12] Ji, G., He, S., Xu, L., Liu, K. and Zhao, J.: Knowledge Graph Embedding via Dynamic Mapping Matrix, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 687–696 (2015).

[13] Nickel, M., Tresp, V. and Kriegel, H.: A Three-Way Model for Collective

Learning on Multi-Relational Data, *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 809–816 (2011).

[14] Yang, B., Yih, W., He, X., Gao, J. and Deng, L.: Embedding Entities and Relations for Learning and Inference in Knowledge Bases, *Proceedings of the Third International Conference on Learning Representations (ICLR)* (2015).

[15] Kazemi, S. M. and Poole, D.: SimplE Embedding for Link Prediction in Knowledge Graphs, *Advances in Neural Information Processing Systems 31*, pp. 4289–4300 (2018).

[16] Socher, R., Chen, D., Manning, C. D. and Ng, A. Y.: Reasoning With Neural Tensor Networks for Knowledge Base Completion, *Advances in Neural Information Processing Systems 26*, pp. 926–934 (2013).

[17] Trouillon, T., Welbl, J., Riedel, S., Gaussier, É. and Bouchard, G.: Complex Embeddings for Simple Link Prediction, *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 2071–2080 (2016).

[18] Nguyen, D. Q.: An overview of embedding models of entities and relationships for knowledge base completion, *CoRR*, Vol. abs/1703.08098 (2017).

[19] Chen, M., Tian, Y., Yang, M. and Zaniolo, C.: Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment, *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, ijcai.org, pp. 1511–1517 (2017).

[20] Sun, Z., Hu, W. and Li, C.: Cross-Lingual Entity Alignment via Joint Attribute-Preserving Embedding, *Proceedings of the 16th International Semantic Web Conference (ISWC)*, pp. 628–644 (2017).

[21] Sun, Z., Hu, W., Zhang, Q. and Qu, Y.: Bootstrapping Entity Alignment with Knowledge Graph Embedding, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4396–4402 (2018).

[22] Mao, X., Wang, W., Xu, H., Lan, M. and Wu, Y.: MRAEA: An Efficient and Robust Entity Alignment Approach for Cross-lingual Knowledge Graph, *Proceedings of the 13th ACM International Conference on Web Search*

and Data Mining (WSDM), pp. 420–428 (2020).

[23] Cao, Y., Liu, Z., Li, C., Liu, Z., Li, J. and Chua, T.: Multi-Channel Graph Neural Network for Entity Alignment, *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pp. 1452–1461 (2019).

[24] Fey, M., Lenssen, J. E., Morris, C., Masci, J. and Kriege, N. M.: Deep Graph Matching Consensus, *Proceedings of th 8th International Conference on Learning Representations (ICLR)* (2020).

[25] Baktashmotlagh, M., Harandi, M. T. and Salzmann, M.: Distribution-Matching Embedding for Visual Domain Adaptation, *Journal of Machine Learning Research*, Vol. 17, pp. 108:1–108:30 (2016).

[26] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K. and Darrell, T.: Deep Domain Confusion: Maximizing for Domain Invariance, *CoRR*, Vol. abs/1412.3474 (2014).

[27] Peyré, G., Cuturi, M. and Solomon, J.: Gromov-Wasserstein Averaging of Kernel and Distance Matrices, *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pp. 2664–2672 (2016).

[28] Nickel, M., Murphy, K., Tresp, V. and Gabrilovich, E.: A Review of Relational Machine Learning for Knowledge Graphs, *Proceedings of the IEEE*, p. 11–33 (2016).

[29] Cuturi, M.: Sinkhorn Distances: Lightspeed Computation of Optimal Transport, *Advances in Neural Information Processing Systems 26*, pp. 2292–2300 (2013).

[30] Peyré, G. and Cuturi, M.: Computational Optimal Transport, *Foundations and Trends in Machine Learning*, Vol. 11, No. 5-6, pp. 355–607 (2019).

[31] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A. C.: Improved Training of Wasserstein GANs, *Advances in Neural Information Processing Systems*, Vol. 30 (2017).

[32] Makkuva, A., Taghvaei, A., Oh, S. and Lee, J.: Optimal transport mapping via input convex neural networks, *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119, pp. 6672–6681 (2020).

[33] Fan, J., Taghvaei, A. and Chen, Y.: Scalable Computations of Wasserstein Barycenter via Input Convex Neural Networks, *CoRR*,

Vol. abs/2007.04462 (2020).

[34] Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P. and Gamon, M.: Representing Text for Joint Embedding of Text and Knowledge Bases, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1499–1509 (2015).

[35] Dettmers, T., Minervini, P., Stenetorp, P. and Riedel, S.: Convolutional 2D Knowledge Graph Embeddings, *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1811–1818 (2018).

[36] Cao, Y., Wang, X., He, X., Hu, Z. and Chua, T.: Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences, *Proceedings of the World Wide Web Conference (WWW)*, pp. 151–161 (2019).

[37] Akiba, T., Sano, S., Yanase, T., Ohta, T. and Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 2623–2631 (2019).

[38] Sansone, E., Ali, H. T. and Sun, J.: Coulomb Autoencoders, *ECAI 2020 - 24th European Conference on Artificial Intelligence*, Vol. 325, pp. 1443–1450 (2020).