SelfKG: Self-Supervised Entity Alignment in Knowledge Graphs

Xiao Liu[†], Haoyun Hong[†], Xinghao Wang[†], Zeyi Chen[†], Evgeny Kharlamov[‡], Yuxiao Dong[†], Jie Tang[†]

[†] Department of Computer Science and Technology, Tsinghua University, China [‡] Bosch Center for AI

{liuxiao21,honghy17,xinghao-18,chenzeyi19}@mails.tsinghua.edu.cn

evgeny.kharlamov@de.bosch.com,{yuxiaod,jietang}@tsinghua.edu.cn

ABSTRACT

Entity alignment, aiming to identify equivalent entities across different knowledge graphs (KGs), is a fundamental problem for constructing Web-scale KGs. Over the course of its development, the label supervision has been considered necessary for accurate alignments. Inspired by the recent progress of self-supervised learning, we explore the extent to which we can get rid of supervision for entity alignment. Commonly, the label information (positive entity pairs) is used to supervise the process of pulling the aligned entities in each positive pair closer. However, our theoretical analysis suggests that the learning of entity alignment can actually benefit more from pushing unlabeled negative pairs far away from each other than pulling labeled positive pairs close. By leveraging this discovery, we develop the self-supervised learning objective for entity alignment. We present SelfKG with efficient strategies to optimize this objective for aligning entities without label supervision. Extensive experiments on benchmark datasets demonstrate that SelfKG without supervision can match or achieve comparable results with state-of-the-art supervised baselines. The performance of SelfKG suggests that self-supervised learning offers great potential for entity alignment in KGs. The code and data are available at https://github.com/THUDM/SelfKG.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Information systems** → **Information integration**.

KEYWORDS

Knowledge Graphs, Self-Supervised Learning, Entity Alignment

ACM Reference Format:

Xiao Liu, Haoyun Hong, Xinghao Wang, Zeyi Chen, Evgeny Kharlamov, Yuxiao Dong, Jie Tang. 2022. SelfKG: Self-Supervised Entity Alignment in Knowledge Graphs. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3511945

Xiao and Haoyun contributed equally to this work.
Jie Tang is the corresponding author.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France.

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3511945

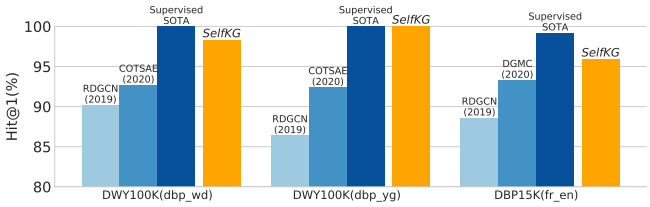


Figure 1: Hit@1 on DWY100K and DBP15K for SelfKG (0% of training labels) and SOTA supervised (100% of training labels) entity alignment. Without using any labels, the self-supervised SelfKG outperforms most of supervised models.

1 INTRODUCTION

Knowledge graphs (KGs) have found widespread adoption in various Web applications, such as search [8, 24], recommendation [12, 19], and question answering [17, 46]. Constructing large-scale KGs has been a very challenging task. While we can extract new facts from scratch, aligning existing (incomplete) KGs together is practically necessary for real-world application scenarios. Over the past years, the problem of entity alignment [35, 39], or namely ontology mapping [20] and schema matching [21], has been a fundamental problem for the Web research community.

Recently, the representation learning-based alignment methods [4, 35, 39, 40, 49] have emerged as the mainstream solutions for entity alignment due to their superior flexibility and accuracy. However, their success relies heavily on the supervision provided by human labeling, which can be biased and arduously expensive to obtain for Web-scale KGs. In light of this fundamental challenge, we aim to explore the potential to align entities across KGs without label supervision (i.e., self-supervised entity alignment).

To achieve this, we revisit the common process of the established supervised entity alignment approaches. Conceptually, for each paired entities from two KGs, the goal of the existing learning objectives is to make them more similar to each other if they are actually the same entity (i.e., a positive pair), otherwise dissimilar if they are different entities (i.e., a negative pair). In the embedding space, this goal is pursued by pulling aligned entities closer and pushing different entities farther away.

We identify the parts where supervision is required in this process. At first place, the supervision serves to pull aligned entities closer. Secondly, another issue arises is the procedure of generating label-aware negative pairs. For every entity in a KG, in the training its negative pairs are formed by randomly sampling entities from the other KG while excluding the groundtruth. If without supervision, it is likely that the implicitly aligned entities are sampled as negative pairs, thus spoiling the training (i.e., collision).

Contributions. We introduce the problem of self-supervised [23] entity alignment in KGs. To address it, we present the SelfKG framework, which does not rely on labeled entity pairs to align entities. It consists of three technical components: 1) relative similarity metric, 2) self-negative sampling, and 3) multiple negative queues.

To get rid of label supervision, we theoretically develop the concept of relative similarity metric (RSM), which enables the self-supervised learning objective. The core idea of RSM is that instead of directly pulling the aligned entities closer in the embedding space, it attempts to push not-aligned negatives far away, thus avoiding the usage of the supervision of positive pairs. In a relative sense, the (implicitly) aligned entities can be considered to be dragged together when optimizing for RSM.

By design, to address the dilemma between supervision with label-aware negative sampling and collision of false-negative samples without it, SelfKG further propose the self-negative sampling strategy, that is, for every entity in a KG, we form its negative pairs by directly sampling entities from the same KG. In other words, SelfKG solely relies on negative entity pairs that are randomly sampled from the input KGs . We theoretically show that this strategy remains effective for aligning entities across KGs.

Finally, our theoretical analysis also shows that the self-supervised loss' error term decays faster as the number of negative samples increases, i.e., a large number of negative samples can benefit SelfKG. However, encoding massive negative samples on the fly is computationally very expensive. We address this by extending the MoCo technique [16] to support two negative queues, each of which corresponds to the two KGs for alignments, ensuring an efficient increase of negative samples.

Empirically, we conduct extensive experiments to demonstrate the premise of self-supervised entity alignment in KGs. We compare the proposed SelfKG method against 24 supervised and one unsupervised baselines on two widely-used entity alignment benchmarks datasets—DWY100K and DBP15K. The results suggest that SelfKG without using any labels can match or achieve comparable performance with the state-of-the-art supervised baselines (Cf. Figure 1). This demonstrates the power of self-supervised learning for entity alignment as well as our design choices of SelfKG.

2 PROBLEM DEFINITION

We introduce the problem of entity alignment in KGs. Conceptually, a KG can be represented as a set of triples T, each of which denotes the relation $r_{ij} \in R$ between two entities $x_i \in E$ and $x_j \in E$. In this work, we denote a KG as $G = \{E, R, T\}$ where E, R, and T are its entity set, relation set, and triple set, respectively.

Given two KGs, $G_x = \{E_x, R_x, T_x\}$ and $G_y = \{E_y, R_y, T_y\}$, the set of the existing aligned entity pairs is defined as $S = \{(x, y)|x \in E_x, y \in E_y, x \Leftrightarrow y\}$, where $\Leftrightarrow$ represents equivalence. The goal of entity alignment between G_x and G_y is to find the equivalent entity from E_x for each entity in E_y, if existed.

Recently, a significant line of work has been focusing on embedding-based techniques for aligning entities in the vector space, e.g., training a neural encoder f to project each entity $x \in E$ into a latent space. Among these attempts, most of them focus on the (semi-) supervised setting in the sense that part of S is used for training the alignment models [4, 35, 39, 40, 49]. Due to the limited alignment labels across KGs in the real world, we instead propose to study to what extent the entity alignment task can be solved in an unsupervised or self-supervised setting, under which none of the existing alignments in S is available.

3 SELF-SUPERVISED ENTITY ALIGNMENT

In this section, we discuss the role that the supervision plays in entity alignment and then present the strategies that can help align entities without label supervision. To this end, we present the SelfKG framework for self-supervised entity alignment across KGs.

3.1 The SelfKG Framework

To enable learning without label information, the main goal of SelfKG is to design a self-supervised objective that can guide its learning process. To achieve this, we propose the concept of *relative similarity metric* (Cf. Section 3.2) between entities across two KGs. To further improve the self-supervised optimization of SelfKG, we introduce the techniques of *self-negative sampling* (Cf. Section 3.3) and *multiple negative queues* (Cf. Section 3.4).

Next, we introduce the initialization of entity embeddings in SelfKG, which is largely built upon existing techniques, including the uni-space learning and GNN based neighborhood aggregator.

Uni-space learning. The idea of uni-space learning has been adopted by recent (semi-) supervised entity alignment techniques [4, 35, 39, 40, 49]. Herein, we present how we leverage it for supporting SelfKG's self-supervised learning setting.

Straightforwardly, embedding entities from different KGs into a uni-space can greatly benefit the alignment task. With labeled entity pairs, it is natural to leverage supervision to align different spaces into one, e.g., merging aligned entities for training [15], or learning projection matrices with abundant training labels to project entities from different embedding spaces into a uni-space [4, 30].

In terms of multi-lingual datasets (e.g., DBP15K), the issue is more challenging. Thanks to the pre-trained language models [14], high-quality multi-lingual initial embeddings are now available. For example, the multi-lingual BERT has been used in recent work [35, 52]. In SelfKG, we adopt LaBSE [9]—a state-of-the-art multi-lingual pre-trained language model trained on 109 different languages—for embedding different knowledge graphs into a uni-space.

Neighborhood aggregator. To further improve the entity embeddings, the neighborhood aggregation is used to aggregate neighbor entities' information to the center entity [39, 42]. In this work, we directly use a single-head graph attention network [37] with one layer to aggregate pre-trained embeddings of one-hop neighbors.

Note that leveraging multi-hop graph structures has been recently explored for the problem of entity alignment. Though some studies [10, 39, 40] claim that they benefit from multi-hop neighbors, other works [42, 51] argue that one-hop neighbors provides enough information for most situations. In our ablation study (Cf. Section 4.2), we find that the multi-hop information actually harms the performance of SelfKG, which is probably resulted from the distant neighbor noises that may be unignorable in a self-supervised setting. Therefore, to demonstrate the minimum requirement of self-supervision for entity alignment, we only involve one-hop neighbor entities during the aggregation.

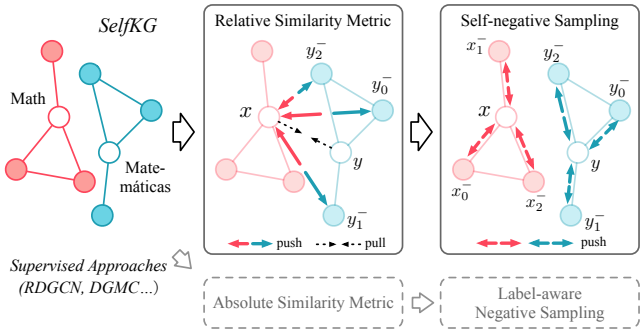


Figure 2: A conceptual comparison of SelfKG and supervised approaches. SelfKG employs the relative similarity metric (RSM) and self-negative sampling to avoid the use of supervision.

3.2 Relative Similarity Metric

We present the self-supervised loss for entity alignment across KGs. First, we analyze the supervised NCE loss for entity alignment. Then, we introduce the relative similarity metric for avoiding labeled pairs. We finally derive the self-supervised NCE for SelfKG.

In representation learning, the margin loss [1, 35] and cross-entropy loss [50] have been widely adopted as the similarity metric. Without loss of generality, they can be expressed in the form of Noise Contrastive Estimation (NCE) [13].

In the context of entity alignment, the NCE loss can be formalized as follows. Let p_x, p_y be the distributions of two KGs G_x, G_y, and p_{pos} denote the representation distribution of the positive entity pairs $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$. Given a pair of aligned entities $(x, y) \sim p_{\text{pos}}$, negative samples $\{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_y$, the temperature τ, and the encoder f satisfies $\|f(\cdot)\| = 1$, we have the supervised NCE loss as

$$\begin{aligned}\mathcal{L}_{\text{NCE}} &\triangleq -\log \frac{e^{f(x)^\top f(y)/\tau}}{e^{f(x)^\top f(y)/\tau} + \sum_i e^{f(x)^\top f(y_i^-)/\tau}} \\ &= \underbrace{-\frac{1}{\tau} f(x)^\top f(y)}_{\text{alignment}} + \underbrace{\log\left(e^{f(x)^\top f(y)/\tau} + \sum_i e^{f(x)^\top f(y_i^-)/\tau}\right)}_{\text{uniformity}}. \end{aligned} \tag{1}$$

where the "alignment" term is to draw the positive pair close and the "uniformity" term is to push the negative pairs away.

We illustrate how this NCE loss can be further adjusted for a self-supervised setting. An example of "pulling" and "pushing" entity pairs in KGs can be found in Figure 2 (left). Previous studies have shown that the NCE loss has the following asymptotic properties:

Theorem 1. (Absolute similarity metric (ASM) [38]) For a fixed $\tau > 0$, as the number of negative samples $M \to \infty$, the (normalized) contrastive loss $\mathcal{L}_{\text{NCE}}$ (i.e., $\mathcal{L}_{\text{ASM}}$) converges to its limit with an absolute deviation decaying in $O(M^{-2/3})$. If a perfectly-uniform encoder f exists, it forms the exact minimizer of the uniformity term.

Proof. Please refer to [38]. □

Theorem 1 makes the NCE loss an absolute similarity metric that requires supervision. However, note that despite potential ambiguity and heterogeneity for entities in KGs, the aligned pairs should share similar semantic meanings, if not exactly the name. Furthermore, the pre-trained word embeddings are known to capture this semantic similarity by projecting similar entities close in the embedding space, which can thus ensure a relatively large $f(x)^T f(y)$ in Eq. 1, i.e., the "alignment" term.

Therefore, to optimize the NCE loss, the main task is then to optimize the "uniformity" term in Eq. 1 rather than the "alignment" term. Considering the boundedness property of f, we can instantly draw an unsupervised upper bound of $\mathcal{L}_{\text{ASM}}$ by as follows.

Proposition 1. Relative similarity metric (RSM). For a fixed $\tau > 0$ and encoder f satisfies $\|f(\cdot)\| = 1$, we always have the following relative similarity metric plus an absolute deviation controlled by a constant as an upper bound for $\mathcal{L}_{\text{ASM}}$:

$$\begin{aligned}\mathcal{L}_{\text{RSM}} &= -\frac{1}{\tau} + \mathop{\mathbb{E}}_{\{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_y} \left[\log\left(e^{1/\tau} + \sum_i e^{f(x)^\top f(y_i^-)/\tau}\right)\right] \\ &\le \mathcal{L}_{\text{ASM}} \le \mathcal{L}_{\text{RSM}} + \frac{1}{\tau}\left[1 - \min_{(x,y)\sim p_{\text{pos}}} \left(f(x)^\top f(y)\right)\right]. \end{aligned} \tag{2}$$

Proof. Please refer to Appendix A.1. □

By optimizing $\mathcal{L}_{\text{RSM}}$, the aligned entities are relatively drawn close by pushing non-aligned ones farther away. In other words, if we cannot draw the aligned entities close (e.g., no positive labels), we can instead push those not-aligned ones far away enough.

By analyzing the commonly-used NCE loss for entity alignment, we find that the training can benefit more from pushing those randomly-sampled (negative) pairs far away than pulling aligned (positive) ones close. Thus, in SelfKG, we focus only on attempting to pushing the negatives far away such that we can get rid of the usage of positive data (i.e., labels).

3.3 Self-Negative Sampling

In the analysis above, we demonstrate that to align entities without supervision, the focus of SelfKG is on sampling negative entity pairs—one from KG G_x and the other from KG G_y. During negative sampling, without supervision for label-aware negative sampling, it is likely that the underlyingly aligned entity pair is sampled as a negative one, i.e., collision happens. Normally, this collision probability can be ignored if a few negatives are sampled; but we discover that a large number of negative samples can be crucial to the success of SelfKG (Cf. Figure 4), under which the collision probability is non-negligible (Cf. Table 4), causing a performance drop by up to 7.7% relatively. To mitigate the issue, we propose to sample negatives x_i^- from G_x for entity $x \in G_x$, given that we are learning from the uni-space of G_x and G_y. By doing so, we would avoid the conflict by simply excluding x, namely self-negative sampling.

However, there may be two other issues aroused consequently. First, due to the real-world noisy data quality, there may often exist several duplicated x in G_x, which could be possibly sampled as negatives. Note that this is also a challenge faced by the supervised setting, where a few duplicated y may also exist in G_y. By following the outline of proof in [38], we show that a certain amount of noise will not influence the convergence of the NCE loss.

Theorem 2. (Noisy ASM) Let the average duplication factors $\lambda \in \mathbb{N}^+, \tau \in \mathbb{R}^+$ be constants. The noisy ASM is denoted as follows and it still converges to the same limit of ASM with the absolute deviation decaying in $O(M^{-2/3})$.

$$\mathcal{L}_{\text{ASM}|\lambda,\text{x}}(f;\tau,M,p_{\text{y}}) = \mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\text{pos}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\text{y}}}} \left[-\log \frac{e^{f(x)^\top f(y)/\tau}}{\lambda e^{f(x)^\top f(y)/\tau} + \sum_i e^{f(x)^\top f(y_i^-)/\tau}} \right] \tag{3}$$

Proof. Please refer to Appendix A.2. □

The second issue is that by changing the negative samples from $y_i^- \in G_y$ to $x_i^- \in G_x$, we need to confirm whether the $\mathcal{L}_{\text{RSM}}$ would still be effective for entity alignment. Empirically, for a selected negative sample $y_j^- \in G_y$, we can expect there to be some partially similar $x_i^- \in G_x$. Since the encoder f is shared for G_x and G_y, the optimization of $f(x_i^-)$ will also contribute to the optimization of $f(y_j^-)$. To justify this, we provide the following theorem.

Theorem 3. (Noisy RSM with self-negative sampling) Let Ω_{x}, Ω_{y} be the spaces of KG triples, respectively, $\{x_i^- : \Omega_{\text{x}} \to \mathbb{R}^n\}_{i=1}^M$, $\{y_i^- : \Omega_{\text{y}} \to \mathbb{R}^n\}_{i=1}^M$ be i.i.d random variables with distribution p_{x}, p_{y}, respectively, and $\mathcal{S}^{d-1}$ denote the uni-sphere in $\mathbb{R}^n$. If there exists a random variable $f : \mathbb{R}^n \to \mathcal{S}^{d-1}$ s.t. $f(x_i^-)$ and $f(y_i^-)$ satisfy the same distribution on $\mathcal{S}^{d-1}$, $1 \le i \le M$., we then have:

$$\lim_{M\to\infty} |\mathcal{L}_{\text{RSM}|\lambda,\text{x}}(f;\tau,M,p_{\text{x}}) - \mathcal{L}_{\text{RSM}|\lambda,\text{x}}(f;\tau,M,p_{\text{y}})| = 0. \tag{4}$$

Proof. Please refer to Appendix A.3. □

Wang et al. [38] suggests that under the condition of $p_{\text{x}} = p_{\text{y}}$, the encoder f can be attained approximately as the minimizer of the uniform loss. Specifically, f follows the uniform distribution on the hypersphere. In SelfKG, the uni-space learning condition ensures the ultimate unified representation for both KGs. The initial p_x and p_y are similar but not identical, which indicates that the self-negative sampling is essential. However, as the training continues, the encoder will be improved as Theorem 2 guarantees to make two KGs more aligned. In other words, the entity embeddings of G_x and G_y could be viewed as the samples from one single distribution in a larger space, i.e., $p_{\text{x}} = p_{\text{y}}$. This in turn allows the existence of f to be more realizable.

In practice, we jointly optimize the loss on both G_x and G_y as follows, which is also illustrated in Figures 2 (right) and 3.

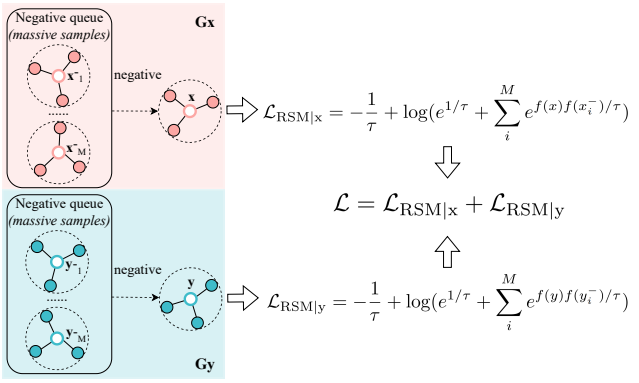
$$\mathcal{L} = \mathcal{L}_{\text{RSM}|\lambda,\text{x}}(f;\tau,M,p_{\text{x}}) + \mathcal{L}_{\text{RSM}|\lambda,\text{y}}(f;\tau,M,p_{\text{y}}). \tag{5}$$

In addition, as the error term of $\mathcal{L}_\lambda(f;\tau,M,p_{\text{x}})$ decays in $O(M^{-2/3})$ (Cf. Theorem 2), we use a comparatively large number of negative samples to boost the performance.

3.4 Multiple Negative Queues

Enlarging the number of negative samples can naturally result in additional computational cost, as encoding massive negative samples on the fly is quite expensive. To address this issue, we propose to extend the MoCo technique [16] for SelfKG. In Moco, a negative queue is maintained to store the previously-encoded batches as the encoded negative samples, which host thousands of encoded negative samples at limited cost.

To adapt to the self-negative sampling strategy in SelfKG, we practically maintain two negative queues, associating with the two input KGs, respectively. An illustrative example is shown in Figure 3. In the beginning, we would not implement the gradient update until one of the queues reaches the predefined length $1+K$ where '1' is for the current batch and K is for the number of previous batches



$$\mathcal{L}_{\text{RSM}|\text{x}} = -\frac{1}{\tau} + \log(e^{1/\tau} + \sum_i^M e^{f(x)f(x_i^-)/\tau})$$

$$\mathcal{L} = \mathcal{L}_{\text{RSM}|\text{x}} + \mathcal{L}_{\text{RSM}|\text{y}}$$

$$\mathcal{L}_{\text{RSM}|\text{y}} = -\frac{1}{\tau} + \log(e^{1/\tau} + \sum_i^M e^{f(y)f(y_i^-)/\tau})$$

Figure 3: The training process of SelfKG. It leverages a negative queue for each KG to provide massive negative samples (up to 4k at a time) for calculating the self-supervised contrastive loss.

used as negative samples. Given $|E|$ as the number of entities in a KG, K, and the batch size N are constraint by

$$(1+K) \times N < \min(|E_x|, |E_y|), \tag{6}$$

it is guaranteed that we would not sample out entities in the current batch. As a result, the real number of negative samples used for the current batch is $(1+K) \times N - 1$.

Momentum update [16]. The main challenge brought by negative queues is the obsolete encoded samples, especially for those encoded at the early stage of training, during which the model parameters vary drastically. Thus, the end-to-end training, which only uses one frequently-updated encoder, may actually harm the training. To mitigate this, we adopt the momentum training strategy, which maintains two encoders—the online encoder and the target encoder. While the online encoder's parameter θ_{online} is instantly updated with the backpropagation, the target encoder θ_{target} for encoding the current batch and then pushing into the negative queue is asynchronously updated with momentum by:

$$\theta_{\text{target}} \leftarrow m \cdot \theta_{\text{target}} + (1-m) \cdot \theta_{\text{online}}, m \in [0,1) \tag{7}$$

A proper momentum is not only important for steady training but may also influence the final performance by avoiding representation collapse (Cf. Figure 4). We present a series of related hyper-parameter studies in Section 4.

Summary. We present SelfKG for self-supervised entity alignment. Figure 2 illustrates that: 1. relative similarity metric (RSM) pushes the non-aligned entities (y_0^-, y_1^- and y_2^-) of x far enough, instead of directly pulling underlyingly-aligned y close to x (labeled pairs), enabling learning without label supervision; 2. self-negative sampling samples negative entities for x from G_x to avoid sampling the true y as its negative. Figure 3 illustrates the training of SelfKG. It leverages existing techniques—embeddings from pre-trained language models and neighborhood aggregator—to initialize entity embeddings into a uni-space. The technical contributions of SelfKG lie in:

(1) the design of the self-supervised loss in Eq. 2 enabled by our relative similarity metric (RSM) in KGs;

Table 1: Statistics of DWY100K and DBP15K. About the definition of neighbor similarity, please refer to Section 4. "#Link" is the number of aligned entity pairs. "#Test Link" is the number of aligned pairs for test.

Model	DWY100K		DBP15K		
	dbp_wd	dbp_yg	zh_en	ja_en	fr_en
#Link	99990	100000	15000	15000	15000
#Test Link	69993	70000	10500	10500	10500
neighbor similarity	0.633	0.777	0.418	0.188	0.182

(2) the strategy of self-negative sampling that furthers Eq. 2 into Eq. 5 to avoid false-negative samples;
(3) the extension of MoCo [16] to two negative queues to support an efficient usage of massive negative samples.

4 EXPERIMENT

We evaluate SelfKG on two widely-acknowledged public benchmarks: DWY100K and DBP15K. DWY100K is a monolingual dataset and DBP15K is a multi-lingual dataset.

DWY100K. The DWY100K dataset used here is originally built by [31]. DWY100K consists of two large datasets: $\text{DWY100K}_{\text{dbp\_wd}}$ (DBpedia to Wikidata) and $\text{DWY100K}_{\text{dbp\_yg}}$ (DBpedia to YAGO3). Each dataset contains 100,000 pairs of aligned entities. However, the entity in the "wd" (Wikidata) part of $\text{DWY100K}_{\text{dbp\_wd}}$ are represented by indices (e.g., Q123) instead of URLs containing entity names, and we search their entity names via the Wikidata[1] API for python.

DBP15K. The DBP15K dataset is originally built by [30][2] and translated into English by [42]. The DBP15K consists of three cross-lingual datasets: $\text{DBP15K}_{\text{zh\_en}}$ (Chinese to English), $\text{DBP15K}_{\text{ja\_en}}$ (Japanese to English) and $\text{DBP15K}_{\text{fr\_en}}$ (French to English). All three datasets are created from multi-lingual DBpedia, and each contains 15,000 pairs of aligned entities. We report results on both original and translated version.

The statistics of DWY100K and DBP15K we use in our work are shown in Table 1. Beyond basic information, we also present a study on datasets' average (1-hop) neighbor similarity, which is the ratio of aligned neighbors of a pair of aligned entities, indicating how noisy the neighborhood information is. We observe that DWY100K's neighborhood information is quite useful, while DBP15K's neighborhood information can be very noisy.

Experiment Setup. We follow the original split of DWY100K [31] and DBP15K [30] which are shown in Table 1. For SelfKG, we randomly take out 5% from the original training set as a dev set for early stopping. We use Hit@k ($k = 1, 10$) to evaluate our model's performance as most works do. The similarity score is calculated using the ℓ_2 distance of two entity embeddings. The batch size is set to 64, momentum m is set to 0.9999, temperature τ is set to 0.08, and queue size is set to 64. We use a learning rate of 10^{-6} with Adam on a Ubuntu server with NVIDIA V100 GPUs (32G).

[1]https://pypi.org/project/Wikidata/
[2]https://github.com/nju-websoft/JAPE

Table 2: Results on DWY100K. Bold results are our best result; underline results are best baseline results.

Model	$\text{DWY100K}_{\text{dbp\_wd}}$		$\text{DWY100K}_{\text{dbp\_yg}}$		macro Hit@1
	Hit@1	Hit@10	Hit@1	Hit@10	
Supervised					
MTransE [4]	0.281	0.520	0.252	0.493	0.267
JAPE [30]	0.318	0.589	0.236	0.484	0.277
IPTransE [54]	0.349	0.638	0.297	0.558	0.322
GCN-Align [39]	0.477	-	0.601	-	0.539
MuGNN [2]	0.616	0.897	0.741	0.937	0.679
RSNs [11]	0.656	-	0.711	-	0.684
BootEA [31]	0.748	0.898	0.761	0.894	0.755
NAEA [55]	0.767	0.918	0.779	0.913	0.773
TransEdge [32]	0.788	0.938	0.792	0.936	0.790
RDGCN [40]	0.902	-	0.864	-	0.883
COTSAE [44]	0.927	0.979	0.944	0.987	0.936
BERT-INT [35]	0.992	-	0.999	-	0.996
CEAFF [49]	<u>1.000</u>	-	<u>1.000</u>	-	1.000
Unsupervised & Self-supervised					
MultiKE [52]	0.915	-	0.880	-	0.898
SelfKG	**0.983**	**0.998**	**1.000**	**1.000**	0.992

4.1 Results

In this part, we report the results of SelfKG and baselines on DWY100K and DBP15K. For all the baselines, we take the reported scores from the corresponding papers, or directly from the tables in BERT-INT [35], CEAFF [49] or NAEA [55]. According to the used proportion of the training labels, we categorize all the models into two types:

- Supervised: 100% of the aligned entity links in the training set is leveraged
- Unsupervised & Self-supervised: 0% of the training set is leveraged.

Overall performance on DWY100K. From Table 2, we observe that SelfKG outperforms all the supervised and unsupervised models except for supervised CEAFF [49] and BERT-INT [35]. However, without any supervision, SelfKG only falls behind supervised state-of-the-art CEAFF on $\text{DWY100K}_{\text{dbp\_wd}}$ by a minimal margin of 1.2%. The reason why $\text{DWY100K}_{\text{dbp\_yg}}$ enables SelfKG to achieve such high accuracy is that the names of its aligned entity pairs are of great similarity respectively, which makes this dataset more easier. The inspiring result implies that at least for monolingual datasets like DWY100K, supervision is not quite necessary for entity alignment.

Overall performance on DBP15K. For the DBP15K dataset, we find that different baselines use different versions of DBP15K in implementation. For example, BERT-INT [35] uses the original multi-lingual version built by [30], while some other methods including RDGCN [40] and DGMC [10] uses machine translation (Google translation) to translate non-English datasets (i.e., zh, ja, fr) of DBP15K into English. If DBP15K is translated, it should not be considered as a multi-lingual setting to some extend. For fair comparison, we report SelfKG's results on both settings.

Table 3: Results on DBP15K. Methods marked with "*" use a translated version of DBP15K [42].Bold results are our best result; underline results are best baseline results.

Model	$DBP15K_{zh\_en}$		$DBP15K_{ja\_en}$		$DBP15K_{fr\_en}$		macro
	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1
Supervised							
MTransE [4]	0.308	0.614	0.279	0.575	0.244	0.556	0.277
JAPE [30]	0.412	0.745	0.363	0.685	0.324	0.667	0.366
IPTransE [54]	0.406	0.735	0.367	0.693	0.333	0.685	0.369
GCN-Align [39]	0.413	0.744	0.399	0.745	0.373	0.745	0.395
SEA [25]	0.424	0.796	0.385	0.783	0.400	0.797	0.403
KECG [18]	0.478	0.835	0.490	0.844	0.486	0.851	0.485
MuGNN [2]	0.494	0.844	0.501	0.857	0.495	0.870	0.497
RSNs [11]	0.508	0.745	0.507	0.737	0.516	0.768	0.510
AliNet [33]	0.539	0.826	0.549	0.831	0.552	0.852	0.547
BootEA [31]	0.629	0.848	0.622	0.854	0.653	0.874	0.635
NAEA [55]	0.650	0.867	0.641	0.873	0.673	0.894	0.655
MRPEA [29]	0.681	0.867	0.655	0.859	0.677	0.890	0.671
JarKA [3]	0.706	0.878	0.646	0.855	0.704	0.888	0.685
TransEdge [32]	0.735	0.919	0.719	0.932	0.710	0.941	0.721
GM-Align [42]	0.679	0.785	0.740	0.872	0.894	0.952	0.771
JAPE [30] *	0.731	0.904	0.828	0.947	-	-	0.780
RDGCN [40] *	0.708	0.846	0.767	0.895	0.886	0.957	0.787
HGCN [41] *	0.720	0.857	0.766	0.897	0.892	0.961	0.793
DGMC [10] *	0.801	0.875	0.848	0.897	0.933	0.960	0.861
RNM [56]*	0.840	0.919	0.872	0.944	0.938	0.954	0.883
CEAFF [49]	0.795	-	0.860	-	0.964	-	0.873
HMAN [43]	0.871	0.987	0.935	0.994	0.973	0.998	0.926
BERT-INT [35]	<u>0.968</u>	<u>0.990</u>	<u>0.964</u>	<u>0.991</u>	<u>0.992</u>	<u>0.998</u>	0.975
Unsupervised & Self-supervised							
MultiKE [52]	0.509	0.576	0.393	0.489	0.639	0.712	0.514
SelfKG	**0.745**	**0.866**	**0.816**	**0.913**	**0.957**	**0.992**	0.840
SelfKG *	0.829	0.919	0.890	0.953	0.959	0.992	0.892

We observe that SelfKG beats all previous supervised ones except for HMAN [43], CEAFF [49] and BERT-INT [35]. There is a gap between supervised state-of-the-arts and SelfKG, which indicates that multi-lingual alignment is surely more complicated than the monolingual setting. We also observe a clear gap between different language datasets. $DBP15K_{zh\_en}$ is the one with the lowest Hit@1, $DBP15K_{ja\_en}$ is the middle, and $DBP15K_{fr\_en}$ has the highest score. However, if we recall the neighbor similarity scores presented in Table 1, it is the $DBP15K_{zh\_en}$ that has the highest neighbor similarity. This discovery indicates that the difference in performance can be mostly attributed to challenges brought by multi-lingual setting instead of structural similarities.

4.2 Ablation Study

We conduct extensive ablation studies respectively on DWY100K and DBP15K for SelfKG. We ablate components regarding the different types of information it brings in. In addition, we conduct studies over some important hyper-parameters using $DBP15K_{zh\_en}$ dataset as an example.

In Table 4, we present the ablation study for SelfKG on both DWY100K and DBP15K, including ablation of neighborhood aggregator and ablation of the self-supervised contrastive training objective based on relative similarity metric (RSM) (i.e., use the original encoding outputs from the LaBSE). We first observe that the LaBSE provides rather good initialization. However, merely the LaBSE is not enough. As we can see, on DWY100K, the LaBSE is benefited substantially from our RSM, with an absolute gain over 10% on $DWY100K_{dbp\_wd}$ and 5% on DBP15K. The use of neighborhood aggregator boosts SelfKG on both DWY100K and DBP15K, which indicates the importance of introducing neighbor information.

Besides, we test the performance of SelfKG without self-negative sampling strategy, which means we sample negative entities from the target KG as most baselines do but without labels (which may introduce the true positive ones). The results show that self-negative sampling is necessary for SelfKG, which brings absolute gains of 2-7%. While the strategy increase in performance can be partly attributed to avoid of collision, careful readers may think of why possibly-existed duplicated entities does not harm as much as the collision. It can be potentially explained that the entity alignment task evaluates alignment accuracy across different KGs (e.g., G_x and G_y) rather than within one KG (e.g., G_x). Even though we might sample duplicated entities in G_x and push them away, it might generate only limited influence on their similarities with the target entity y in G_y.

Impact of the quality of pre-trained uni-space embedding. To clarify the influence of different pre-trained word embeddings, we conduct an experiment that replaces the LaBSE embedding we use in SelfKG with FastText embeddings, which is widely used in baseline methods.

First, comparing FastText results with and without training, the after-training results are consistently higher by 8.5%-17.2% than before-training results in Table 5. These results also outperform all previous unsupervised baselines, indicating the effectiveness of SelfKG when being applied to any embedding initialization.

Second, comparing FastText results with LaBSE results, we also confirm that a stronger pre-trained language model like LaBSE will boost SelfKG's performance compared to FastText word embeddings. This is also the case in baseline methods, such as HMAN [43] and BERT-INT [35], who leverage multi-lingual BERT as their encoders. Despite better pre-trained embeddings, in our ablation study (Cf. Table 4 and Table 5), we show that the "-w.o. RSM + neighbors" (i.e. LaBSE before SelfKG training) can be significantly improved by 6.4%-28.2% with SelfKG, which demonstrates the usefulness of our method.

Impact of relation information and multi-hop structure information. To better examine whether relational structural information will help in the self-supervised setting (which might have different results from previous supervised observations), we first conduct experiments on incorporating multi-hop information and then integrate relation information. Table 6 shows the results on DBP15K when multi-hop neighbors (more specifically, 20-nearest-neighbor subgraph) are leveraged instead of 1-hop ones. We observe that the performance is actually worse. This is probably because of the heterogeneity of different knowledge graphs and also because the neighbor noises may be amplified in a self-supervised setting.

Table 4: Ablation Study of SelfKG's components and strategies on DWY100K and DBP15K.

Model	$DWY100K_{dbp\_wd}$		$DWY100K_{dbp\_yg}$		$DBP15K_{zh\_en}$		$DBP15K_{ja\_en}$		$DBP15K_{fr\_en}$	
	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10
SelfKG	**0.983**	**0.998**	**1.000**	**1.000**	**0.745**	**0.866**	**0.816**	**0.913**	**0.957**	**0.992**
-w.o. RSM	0.884	0.963	**1.000**	**1.000**	0.670	0.813	0.760	0.867	0.916	0.987
-w.o. neighbors	0.887	0.987	**1.000**	**1.000**	0.638	0.783	0.732	0.849	0.931	0.978
-w.o. RSM + neighbors	0.799	0.903	**1.000**	**1.000**	0.581	0.739	0.689	0.815	0.899	0.964
-w.o. self negative sampling	0.918	0.978	**1.000**	**1.000**	0.688	0.833	0.773	0.882	0.932	0.980

Table 5: Ablation Study on quality of pre-trained uni-space embedding on DWY100K and DBP15K.

Model	$DWY100K_{dbp\_wd}$		$DWY100K_{dbp\_yg}$		$DBP15K_{zh\_en}$		$DBP15K_{ja\_en}$		$DBP15K_{fr\_en}$	
	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10
FastText - before SelfKG training	0.837	0.910	0.864	0.939	0.590	0.688	0.645	0.755	0.828	0.898
- after SelfKG training	**0.921**	**0.986**	**0.954**	**0.993**	**0.707**	**0.834**	**0.755**	**0.865**	**0.914**	**0.967**
LaBSE - before SelfKG training	0.799	0.903	**1.000**	**1.000**	0.581	0.739	0.689	0.815	0.899	0.964
- after SelfKG training	**0.983**	**0.998**	**1.000**	**1.000**	**0.745**	**0.866**	**0.816**	**0.913**	**0.957**	**0.992**

Table 6: Ablation Study of multi-hop structure and relation information on DBP15K.

Model	$DBP15K_{zh\_en}$		$DBP15K_{ja\_en}$		$DBP15K_{fr\_en}$	
	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10
SelfKG	0.745	0.866	0.816	0.913	0.957	0.992
multi-hop	0.685	0.834	0.769	0.876	0.936	0.983
with relation	**0.750**	**0.876**	**0.819**	**0.921**	**0.959**	**0.994**

Based on the 1-hop restriction, as for incorporating relation information, we combine relation name embeddings and their corresponding tail entity name embeddings as the new 1-hop neighbor embeddings. We can see that the results are improved by a slight margin with relation information, which demonstrates that relational information is of a little usefulness.

Impact of hyper-parameters. The main hyper-parameters in SelfKG are 1) negative queue size and batch size (which influence the capacity of negative samples), and 2) momentum coefficient m that controls SelfKG's training stability.

As pointed out in Theorem 1 and 2, the error term of contrastive loss decays with $O(M^{-2/3})$, which indicates the importance of enlarging the number of negative samples. Fixing batch size to 64, we change the sizes of the negative queue and derive the curve in Figure 4. The performance increase is not obvious when queue size is between 10^0 and 10^1; but as it grows to 10^2, the improvement becomes significant. Fixing queue size to 64, along the increase of batch size, the improvement is more stable ranging from 10^1 to 10^2.

For momentum coefficient m, we discover that a properly large m such as 0.9999 is usually better for SelfKG. Besides, a proper m is also critical for better training stability (Cf. Figure 4). A small momentum leads to faster convergence, but also representation collapse and consequent poorer performance. A too-large momentum (e.g., 0.99999) converges too slow.

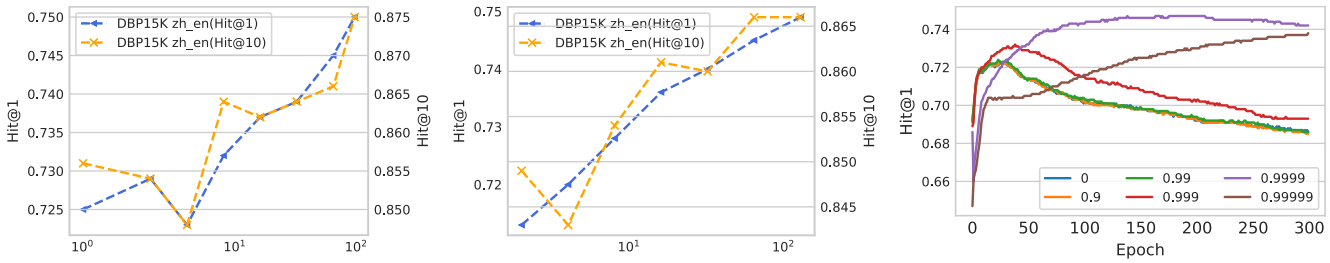
4.3 SelfKG v.s. Supervised SelfKG

In practice, we often encounter low-data resource situations where there is very limited supervision. To justify SelfKG's scalability, we compare self-supervised SelfKG with its supervised counterpart SelfKG (sup) on $DBP15K_{zh\_en}$ across different data resource settings. SelfKG (sup) follows the conventional supervised entity alignment methods using Absolute Similarity Metric as presented in Eq. 3.

In our preliminary experiment, we find that the original DBP15k's data split (30% labels for training and 70% for testing) is not sufficient to present SelfKG (sup)'s advantage, resulting in a Hit@1 of 0.744 for SelfKG (sup) and 0.745 for SelfKG. So we construct a new split of $DBP15K_{zh\_en}$ that contains 20% for testing and 80% for constructing different sizes of training set. The result is presented in Figure 5, where the horizontal axis indicates the ratio of training labeled entities for SelfKG (sup) to all entities. We observe that SelfKG is approximately comparable to SelfKG (sup) using an amount of 25% labeled entities, which accords with our observation in the aforementioned preliminary experiment. When using less than 25% amount of labeled entities, SelfKG performs much better than SelfKG (sup), which demonstrates the effectiveness of SelfKG in low supervised data resource settings.

5 RELATED WORK

Entity alignment. Entity alignment, also named entity resolution, ontology alignment, or schema matching, is a fundamental problem in the knowledge graph community [48] that has been researched for decades. Before the deep learning era, most approaches focus on designing proper similarity factors and Bayesian-based probability estimation. [34] develops the idea of transforming the alignment into minimizing the risk of decision making. RiMOM [20] proposes a multi-strategy ontology alignment framework, which leverages primary similarity factors with the Cartesian product to align concepts unsupervisedly. [21] argues for rule-based linking and design



(a) Negative queue size (b) Batch size (c) Momentum

Figure 4: Study on (a) negative queue size, (b) batch size, and (c) momentum on $DBP15K_{zh\_en}$. (c) presents the test Hit@1 curve throughout the training epochs.

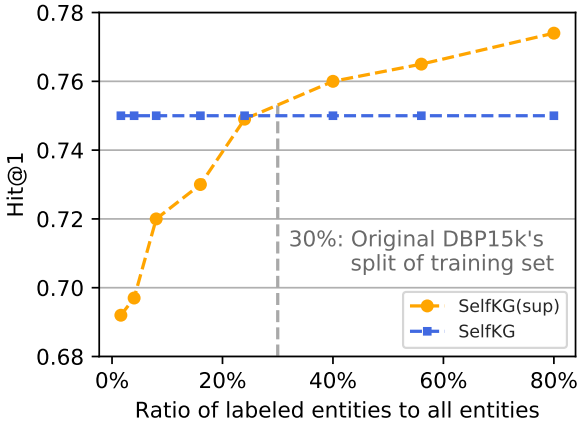


Figure 5: SelfKG vs. SelfKG (sup) on DBP15K. SelfKG works well in a low-data resource setting.

a rule discovery algorithm. [53] develops an efficient multi-network linking algorithm based on the factor graph model.

Recently, embedding-based methods have drawn people's attention due to their flexibility and effectiveness. TransE [1] is the very beginning to introduce the embedding method to represent relational data. [4] develops the knowledge graph alignment strategy based on TransE. [30] argues for a cross-lingual entity alignment task and constructs the dataset from DBpedia. [51] proposes to embed entity ego-network to vectors for the alignment. [39] introduces the GCN to model both the entity and relation in knowledge graphs to perform the alignment. [36] argues that we can use attributes and structure to supervise each other mutually. BERT-INT [35] proposes an interactive entity alignment strategy based on BERT and substantially improves the supervised entity alignment performance on public benchmarks. [50] designs heterogeneous graph attention networks to perform large-scale entity linking across the open academic graph.

However, most embedding-based methods nowadays rely heavily on supervised data, hindering their application in real web-scale noisy data. As a prior effort, in [22] authors present self-supervised pre-training for concept linking but with downstream supervised classification. In this work, we endeavor to investigate the potential of a completely self-supervised approach without using labels to reduce the cost of entity alignment while improving performance.

Self-supervised learning. Self-supervised learning [23], which learns the co-occurrence relationships in the data without human supervision, is a data-efficient and powerful machine learning paradigm. We can divide them into two categories: generative and contrastive.

Generative self-supervised learning is often related to pre-training. For instance, BERT [6], GPT [27], XLNet [45] and so on [7, 28] develop the field of language model pre-training, which boost the development of natural language processing. The contrastive self-supervised learning is recently proposed by MoCo and SimCLR [5, 16] in computer vision to conduct successful vision pre-training. The core idea of leveraging the instance discrimination and contrastive loss has been proved to be especially useful for downstream classification tasks. Self-supervised learning has also been applied to graph pre-training tasks, such as in GCC [26], the authors pre-train the structural representation of subgraphs using contrastive learning and transfer the model to other graphs. [47] proposes adding augmentations to sampled graphs following SimCLR's strategy to promote graph pre-training performance.

6 CONCLUSION

In this work, we re-examine the use and effect of supervision in the entity alignment problem, which targets aligning entities with identical meanings across different knowledge graphs. Based on the three insights we derive—uni-space learning, relative similarity metric, and self-negative sampling, we develop a self-supervised entity alignment algorithm—SelfKG—to automatically align entities without training labels. The experiments on two widely-used benchmarks DWY100K and DBP15K show that SelfKG is able to beat or match most of the supervised alignment methods which leverage the 100% of the training datasets. Our discovery indicates a huge potential to get rid of supervision in the entity alignment problem, and more studies are expected for a deeper understanding of self-supervised learning.

ACKNOWLEDGMENTS

The work is supported by the NSFC for Distinguished Young Scholar (61825602), NSFC (61836013), and Tsinghua-Bosch Joint ML Center. Haoyun Hong is supported by Tsinghua University Initiative Scientific Research Program and DCST Student Academic Training Program.

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*. 1–9.

[2] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Juanzi Li, and Tat-Seng Chua. 2019. Multi-Channel Graph Neural Network for Entity Alignment. In *ACL*. 1452–1461.

[3] Bo Chen, Jing Zhang, Xiaobin Tang, Hong Chen, and Cuiping Li. 2020. JarKA: Modeling Attribute Interactions for Cross-lingual Knowledge Alignment. In *PAKDD*. Springer.

[4] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. PMLR, 1597–1607.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.

[7] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. All nlp tasks are generation tasks: A general pretraining framework. *arXiv preprint arXiv:2103.10360* (2021).

[8] Jeffrey Scott Eder. 2012. Knowledge graph based search system. US Patent App. 13/404,109.

[9] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852* (2020).

[10] Matthias Fey, Jan E Lenssen, Christopher Morris, Jonathan Masci, and Nils M Kriege. 2020. Deep Graph Matching Consensus. In *ICLR*.

[11] Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *ICML*. PMLR, 2505–2514.

[12] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *TKDE* (2020).

[13] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AIStats*. 297–304.

[14] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, et al. 2021. Pre-trained models: Past, present and future. *AI Open* (2021).

[15] Yanchao Hao, Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. 2016. A joint embedding method for entity alignment of knowledge bases. In *CCKS*. Springer, 3–14.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.

[17] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NIPS* (2020).

[18] Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. 2019. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In *EMNLP*. 2723–2732.

[19] Feng-Lin Li, Hehong Chen, Guohai Xu, Tian Qiu, Feng Ji, Ji Zhang, and Haiqing Chen. 2020. AliMeKG: Domain Knowledge Graph Construction and Application in E-commerce. In *CIKM*. 2581–2588.

[20] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. 2008. Rimom: A dynamic multistrategy ontology alignment framework. *TKDE* 21, 8 (2008), 1218–1232.

[21] Lingli Li, Jianzhong Li, and Hong Gao. 2014. Rule-based method for entity resolution. *TKDE* 27, 1 (2014), 250–263.

[22] Xiao Liu, Li Mian, Yuxiao Dong, Fanjin Zhang, Jing Zhang, Jie Tang, Peng Zhang, Jibing Gong, and Kuansan Wang. 2021. OAG_know: Self-supervised Learning for Linking Knowledge Graphs. *TKDE* (2021).

[23] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *TKDE* (2021).

[24] Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web* 8, 3 (2017), 489–508.

[25] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *WWW*. 3130–3136.

[26] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *SIGKDD*. 1150–1160.

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[28] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. [n. d.]. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR* ([n. d.]).

[29] Xiaofei Shi and Yanghua Xiao. 2019. Modeling multi-mapping relations for precise cross-lingual entity alignment. In *EMNLP*. 813–822.

[30] Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*. Springer, 628–644.

[31] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In *IJCAI, vol.18*. 4396–4402.

[32] Zequn Sun, Jiacheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In *ISWC*. Springer, 612–629.

[33] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *AAAI*, Vol. 34. 222–229.

[34] Jie Tang, Juanzi Li, Bangyong Liang, Xiaotong Huang, Yi Li, and Kehong Wang. 2006. Using Bayesian decision for ontology mapping. *JWS* 4, 4 (2006), 243–262.

[35] Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. 2021. BERT-INT: a BERT-based interaction model for knowledge graph alignment. In *IJCAI*. 3174–3180.

[36] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity alignment between knowledge graphs using attribute embeddings. In *AAAI*, Vol. 33. 297–304.

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[38] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*. PMLR, 9929–9939.

[39] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*. 349–357.

[40] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs. In *IJCAI*.

[41] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019. Jointly Learning Entity and Relation Representations for Entity Alignment. In *EMNLP*. 240–249.

[42] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. 2019. Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. In *ACL*.

[43] Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. 2019. Aligning Cross-Lingual Entities with Multi-Aspect Information. In *EMNLP*. 4431–4441.

[44] Kai Yang, Shaoqin Liu, Junfeng Zhao, Yasha Wang, and Bing Xie. 2020. COTSAE: CO-Training of Structure and Attribute Embeddings for Entity Alignment. In *AAAI*, Vol. 34. 3025–3032.

[45] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *NIPS* (2019).

[46] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *EMNLP*.

[47] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *NIPS* (2020).

[48] Kaisheng Zeng, Chengjiang Li, Lei Hou, Juanzi Li, and Ling Feng. 2021. A comprehensive survey of entity alignment for knowledge graphs. *AI Open* 2 (2021), 1–13.

[49] Weixin Zeng, Xiang Zhao, Jiuyang Tang, and Xuemin Lin. 2019. Collective Embedding-based Entity Alignment via Adaptive Features.

[50] Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, et al. 2019. Oag: Toward linking large-scale heterogeneous entity graphs. In *SIGKDD*. 2585–2595.

[51] Jing Zhang, Bo Chen, Xianming Wang, Hong Chen, Cuiping Li, Fengmei Jin, Guojie Song, and Yutao Zhang. 2018. Mego2vec: Embedding matched ego networks for user alignment across social networks. In *CIKM*. 327–336.

[52] Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Multi-view Knowledge Graph Embedding for Entity Alignment. In *IJCAI*. 5429–5435.

[53] Yutao Zhang, Jie Tang, Zhilin Yang, Jian Pei, and Philip S Yu. 2015. Cosnet: Connecting heterogeneous social networks with local and global consistency. In *SIGKDD*. 1485–1494.

[54] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative Entity Alignment via Joint Knowledge Embeddings.. In *IJCAI*, Vol. 17. 4258–4264.

[55] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2019. Neighborhood-Aware Attentional Representation for Multilingual Knowledge Graphs.. In *IJCAI*. 1943–1949.

[56] Yao Zhu, Hongzhi Liu, Zhonghai Wu, and Yingpeng Du. 2021. Relation-Aware Neighborhood Matching Model for Entity Alignment. In *AAAI*, Vol. 35. 4749–4756.

A APPENDIX

A.1 Proof to Proposition 1

Proof. Notice that $\frac{x}{x+a}$ is increasing w.r.t $x \in \mathbb{R}, x \geq 0$, where $a \in \mathbb{R}, a > 0$ is a constant. Then we have:

$$\begin{aligned}\mathcal{L}_{\text{RSM}} &= \mathop{\mathbb{E}}_{\{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}\left[-\log \frac{e^{\frac{1}{\tau}}}{e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}}\right] \\ &\leq \mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\text{pos}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}}\left[-\log \frac{e^{f(x)^{\mathsf{T}} f(y)/\tau}}{e^{f(x)^{\mathsf{T}} f(y)/\tau} + \sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}}\right] = \mathcal{L}_{\text{ASM}}.\end{aligned} \tag{8}$$

On the other hand,

$$\begin{aligned}\mathcal{L}_{\text{ASM}} &\leq \mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\text{pos}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}}\left[-\log\left(\frac{e^{\min\left(f(x)^{\mathsf{T}} f(y)\right)/\tau}}{e^{\min\left(f(x)^{\mathsf{T}} f(y)\right)/\tau} + \sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}}\right)\right] \\ &\leq \mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\text{pos}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}}\left[-\log\left(\frac{e^{\min\left(f(x)^{\mathsf{T}} f(y)\right)/\tau}}{e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}}\right)\right] \\ &\leq \mathcal{L}_{\text{RSM}} + \frac{1}{\tau}\left[1 - \min_{(x,y)\sim p_{\text{pos}}}\left(f(x)^{\mathsf{T}} f(y)\right)\right].\end{aligned} \tag{9}$$

□

A.2 Proof to Theorem 2

Proof. We follow the outline of Wang's proof [38].

$$\begin{aligned}&\lim_{M\to\infty}\left[\mathcal{L}_{\text{ASM}|\lambda,\mathsf{x}}(f;\tau,M,p_{\mathsf{y}}) - \log M\right] \\ &= -\frac{1}{\tau}\mathop{\mathbb{E}}_{(x,y)\sim p_{\text{pos}}}\left[f(x)^{\mathsf{T}} f(y)\right] \\ &+ \lim_{M\to\infty}\mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\text{pos}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}}\left[\log\left(\frac{\lambda}{M}e^{f(x)^{\mathsf{T}} f(y)/\tau} + \frac{1}{M}\sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}\right)\right] \\ &= -\frac{1}{\tau}\mathop{\mathbb{E}}_{(x,y)\sim p_{\text{pos}}}\left[f(x)^{\mathsf{T}} f(y)\right] + \mathop{\mathbb{E}}_{x\overset{\text{i.i.d.}}{\sim} p_{\mathsf{x}}}\left[\log \mathop{\mathbb{E}}_{y^-\overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}\left[e^{f(x)^{\mathsf{T}} f(y^-)/\tau}\right]\right]\end{aligned} \tag{10}$$

where the last equality is by the S.L.L.N. (Strong Law of Large Numbers) and the Continuous Mapping Theorem.

The convergence speed is derived as follows, where $\lambda \geq 1$ and $-1 \leq f(x)^T f(y), f(x)^T f(y_i^-) \leq 1$.

For one side:

$$\begin{aligned}&\mathcal{L}_{\text{ASM}|\lambda,\mathsf{x}}(f;\tau,M,p_{\mathsf{y}}) - \log M - \lim_{M\to\infty}\left[\mathcal{L}_{\text{ASM}|\lambda,\mathsf{x}}(f;\tau,M,p_{\mathsf{y}}) - \log M\right] \\ &\leq \mathop{\mathbb{E}}_{\substack{x\overset{\text{i.i.d.}}{\sim} p_{\mathsf{x}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}}\left[\log\left(\frac{\lambda}{M}e^{1/\tau} + \frac{1}{M}\sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}\right)\right] \\ &- \mathop{\mathbb{E}}_{x\overset{\text{i.i.d.}}{\sim} p_{\mathsf{x}}}\left[\log \mathop{\mathbb{E}}_{y^-\overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}\left[e^{f(x)^{\mathsf{T}} f(y^-)/\tau}\right]\right] \\ &\leq \mathop{\mathbb{E}}_{x\overset{\text{i.i.d.}}{\sim} p_{\mathsf{x}}}\left[\log \mathop{\mathbb{E}}_{y^-\overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}\left[\left(\frac{\lambda}{M}e^{1/\tau} + e^{f(x)^{\mathsf{T}} f(y^-)/\tau}\right)\right] - \log \mathop{\mathbb{E}}_{y^-\overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}\left[e^{f(x)^{\mathsf{T}} f(y^-)/\tau}\right]\right] \\ &\leq \mathop{\mathbb{E}}_{x\overset{\text{i.i.d.}}{\sim} p_{\mathsf{x}}}\left[\frac{\lambda}{M}e^{2/\tau}\right] \\ &= \frac{\lambda}{M}e^{2/\tau},\end{aligned} \tag{11}$$

where the second inequality follows the Jensen Inequality based on the the concavity of log.

For the other side:

$$\begin{aligned}&\lim_{M\to\infty}\left[\mathcal{L}_{\text{ASM}|\lambda,\mathsf{x}}(f;\tau,M,p_{\mathsf{y}}) - \log M\right] - \left[\mathcal{L}_{\text{ASM}|\lambda,\mathsf{x}}(f;\tau,M,p_{\mathsf{y}}) - \log M\right] \\ &\leq e^{1/\tau}\mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\text{pos}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}}\left[\left|\mathop{\mathbb{E}}_{y^-\overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}\left[e^{f(x)^{\mathsf{T}} f(y^-)/\tau}\right] - \left(\frac{\lambda}{M}e^{f(x)^{\mathsf{T}} f(y)/\tau} + \frac{1}{M}\sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}\right)\right|\right] \\ &\leq \frac{\lambda}{M}e^{2/\tau} + e^{1/\tau}\mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\text{pos}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}}\left[\left|\mathop{\mathbb{E}}_{y^-\overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}\left[e^{f(x)^{\mathsf{T}} f(y^-)/\tau}\right] - \frac{1}{M}\sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}\right|\right] \\ &\leq \frac{\lambda}{M}e^{2/\tau} + \frac{5}{4}M^{-\frac{2}{3}}e^{\frac{1}{\tau}}\left(e^{\frac{1}{\tau}} - e^{-\frac{1}{\tau}}\right),\end{aligned} \tag{12}$$

where the first inequality follows an application of Lagrange's mean-value theorem, and the last inequality follows the bound from Chebychev's inequality, which can refer to [38].

Therefore, The noisy ASM still converges to the same limit of ASM with absolute deviation decaying in $O(M^{-2/3})$, combing the derivations of both sides above. □

A.3 Proof to Theorem 3

Let $\Omega_{\mathsf{x}}, \Omega_{\mathsf{y}}$ be the space of knowledge graph triplets, $n \in \mathbb{N}$. Let $\{x_i^- : \Omega_{\mathsf{x}} \to \mathbb{R}^n\}_{i=1}^M$, $\{y_i^- : \Omega_{\mathsf{y}} \to \mathbb{R}^n\}_{i=1}^M$ be i.i.d random variables with distribution $p_{\mathsf{x}}, p_{\mathsf{y}}$. $\mathcal{S}^{d-1}$ denotes the uni-sphere in $\mathbb{R}^n$. If there exists a random variable $f : \mathbb{R}^n \to \mathcal{S}^{d-1}$ s.t. $f(x_i^-), f(y_i^-)$ satisfy the same distribution on $\mathcal{S}^{d-1}, 1 \leq i \leq M.$, then we have

$$\lim_{M\to\infty}\left|\mathcal{L}_{\text{RSM}|\lambda,\mathsf{x}}(f;\tau,M,p_{\mathsf{x}}) - \mathcal{L}_{\text{RSM}|\lambda,\mathsf{x}}(f;\tau,M,p_{\mathsf{y}})\right| = 0. \tag{13}$$

Proof.

$$\begin{aligned}&\left|\mathcal{L}_{\text{RSM}|\lambda,\mathsf{x}}(f;\tau,M,p_{\mathsf{x}}) - \mathcal{L}_{\text{RSM}|\lambda,\mathsf{x}}(f;\tau,M,p_{\mathsf{y}})\right| \\ &= \left|\mathop{\mathbb{E}}_{\{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{x}}}\left[-\log\left(\frac{e^{\frac{1}{\tau}}}{\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(x_i^-)/\tau}}\right)\right]\right. \\ &\quad \left. - \mathop{\mathbb{E}}_{\{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}\left[-\log\left(\frac{e^{\frac{1}{\tau}}}{\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}}\right)\right]\right| \\ &= \left|\mathop{\mathbb{E}}_{\substack{\{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{x}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}}\left[\log\left(\frac{\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(x_i^-)/\tau}}{\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}}\right)\right]\right| \\ &\leq \mathop{\mathbb{E}}_{\substack{\{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{x}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}}\left[\left|\log\left(\frac{\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(x_i^-)/\tau}}{\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}}\right)\right|\right] \\ &= \mathop{\mathbb{E}}_{\substack{\{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{x}} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\mathsf{y}}}}\left[\left|\log\left(1 + \frac{\sum_i e^{f(x)^{\mathsf{T}} f(x_i^-)/\tau} - \sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}}{\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}}\right)\right|\right].\end{aligned} \tag{14}$$

Let $S = \frac{\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(x_i^-)/\tau}}{\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^{\mathsf{T}} f(y_i^-)/\tau}}$, then

$$S \geq \frac{\lambda e^{\frac{1}{\tau}} + M e^{-\frac{1}{\tau}}}{\lambda e^{\frac{1}{\tau}} + M e^{\frac{1}{\tau}}} \geq e^{-\frac{2}{\tau}}. \tag{15}$$

Let $T = \frac{\sum_i e^{f(x)^\top f(x_i^-)/\tau} - \sum_i e^{f(x)^\top f(y_i^-)/\tau}}{\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^\top f(y_i^-)/\tau}}$, therefore

$$|T| = \frac{\frac{1}{M}\left|\sum_i e^{f(x)^\top f(x_i^-)/\tau} - \sum_i e^{f(x)^\top f(y_i^-)/\tau}\right|}{\frac{1}{M}\left(\lambda e^{\frac{1}{\tau}} + \sum_i e^{f(x)^\top f(y_i^-)/\tau}\right)} \leq \frac{e^{\frac{1}{\tau}} - e^{-\frac{1}{\tau}}}{\frac{\lambda}{M} e^{\frac{1}{\tau}} + e^{-\frac{1}{\tau}}}. \quad (16)$$

Then $S = 1 + T$, therefore

$$S \leq 1 + \frac{e^{\frac{1}{\tau}} - e^{-\frac{1}{\tau}}}{\frac{\lambda}{M} e^{\frac{1}{\tau}} + e^{-\frac{1}{\tau}}} < 1 + \frac{e^{\frac{1}{\tau}} - e^{-\frac{1}{\tau}}}{e^{-\frac{1}{\tau}}} = 1 + e^{\frac{2}{\tau}} - 1 = e^{\frac{2}{\tau}}. \quad (17)$$

Therefore, $|\log S| < \frac{2}{\tau}$.

By the S.L.L.N., $\lim_{M\to\infty} T = 0$, therefore, $\lim_{M\to\infty} \log S = 0$.

Because $|\log S|$ is bounded, with Dominated Covergence Theorem, the sign of mathematical expectation (i.e. integral) can be exchanged with the sign of limit:

$$\begin{aligned}
&\lim_{M\to\infty} \left|\mathcal{L}_{\mathrm{RSM}|\lambda,\mathrm{x}}(f;\tau,M,p_\mathrm{x}) - \mathcal{L}_{\mathrm{RSM}|\lambda,\mathrm{x}}(f;\tau,M,p_\mathrm{y})\right| \\
&\leq \lim_{M\to\infty} \mathbb{E}_{\substack{\{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_\mathrm{x} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_\mathrm{y}}} |\log S| \\
&= \mathop{\mathbb{E}}_{\substack{\{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_\mathrm{x} \\ \{y_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_\mathrm{y}}} \left[\lim_{M\to\infty} |\log S|\right] \\
&= 0.
\end{aligned} \quad (18)$$

□

A.4 Details on Implementation

A.4.1 Dataset. For both DWY100K[3] and DBP15K[4] datasets we used, we do simple data processing on the original datasets built in BootEA [31] and JAPE [30] respectively. The process of data processing is as follows:

Firstly, we remove the redundant prefixes of the URLs representing the entities, leaving the meaningful entity names at the end. For example, in $\mathrm{DBP15K_{zh\_en}}$ dataset, there is an entity represented by "http://dbpedia.org/resource/2012_Summer_Olympics". We remove the substring in front of "2012_Summer_Olympics" to remove the useless part. Then we replace the underscores used to connect words in the entity names with spaces, so that the entities can be represented by their original entity names. In addition, we replace the indices that represent entities in $\mathrm{DWY100K_{dbp\_wd}}$ (e.g., Q123) with strings of entity names. The purpose of this step is to make the entity names as original as possible to let our model better extract the character-level information and the semantic-level information with useful data. Then, we need to map every entity to a unique index in every pair of KGs respectively. The pairs of KGs are the subdatasets of DWY100K and DBP15K: $\mathrm{DWY100K_{dbp\_wd}}$, $\mathrm{DWY100K_{dbp\_yg}}$, $\mathrm{DBP15K_{zh\_en}}$, $\mathrm{DBP15K_{ja\_en}}$ and $\mathrm{DBP15K_{fr\_en}}$. We use DBP15K dataset provided in [42] and the DWY100K dataset provided in [30] as our original dataset and follow the indices they created in our experiments since they have already done this processing step.

As for obtaining 1-hop neighbors, we treat the KGs as undirected graphs, that means we use the relational triples in the datasets to find all the entities connected to an entity regardless of the direction of the connection.

Finally, we reconstruct Dataset and use DataLoader of Pytorch's torch.utils.data package to packet our data and create batches. Because we do not use any labels in our model for training, we set the indices of the entities as the y data which is usually used to contain the labels in Dataset package. As for the x data which is the training data in Dataset package, we set the entity names of the center entities and the corresponding neighbors with the adjacency matrix of the center entities as the x data.

A.4.2 Implementation Notes. Our model is implemented using Python package Pytorch 1.7.1.[5] The experiments were conducted on a GNU/Linux server with 8 Tesla V100 SXM2 GPU and 32G GPU RAM mainly, and also 56 Intel(R) Xeon(R) Gold 5120 CPU(2.20GHz), 500G RAM.

For both experiments on DWY100K and DBP15K, we randomly select 5% links from the training set in the original datasets as our validation set and evaluate our model's performance both on the validation set and the testing set. We stop the training progress once our model reaches the best performance on the validation set and record Hit@1 and Hit@10 results on the testing set.

A.4.3 Similarity Search. In order to evaluate our model on the validation set and the test set efficiently, we apply Faiss[6], a library for efficient similarity search.

In the evaluation period, we apply the IndexFlatL2 as indexer, which is based on ℓ_2 distance. Once the indices are built, via the kd-tree algorithm used in Faiss, the top 1 and top 10 closest entities in the target KG of every entity in the source KG can be found efficiently.

A.5 Runtime

On the time efficiency of using large number negative samples in SelfKG, by leveraging multiple negative queues with Moco [16], the running time of SelfKG is significantly reduced even when the sample size is large, making it similar to the common negative sampling method adopted in state-of-the-art baseline methods. Details are discussed in Section 3.4.

A.6 Limitations

There are mainly two limitations in SelfKG. Firstly, SelfKG requires good embeddings to ensure the unified representation for both KGs. As we clarified in Section 4.2, we confirm that a better pre-trained language model like LaBSE will boost the performance of SelfKG. This issue is also commonly faced by other embedding-based entity alignment methods. Secondly, SelfKG still underperforms some supervised state-of-the-art methods. Some of the supervised methods such as BERT-INT[35] can reach almost an accuracy of 100% on both DBP15K and DWY100K, which outperforms our self-supervised solution. The gap is expected since supervision does provide much useful information for the alignment task. The ultimate goal of self-supervised methods is to match or even beat supervised methods.

[3]Can be downloaded from https://github.com/nju-websoft/BootEA
[4]Can be downloaded from https://github.com/syxu828/Crosslingula-KG-Matching
[5]More details can be found in our code https://github.com/THUDM/SelfKG
[6]https://github.com/facebookresearch/faiss