LAMBDA: Learning Matchable Prior For Entity Alignment with Unlabeled Dangling Cases

Hang Yin, Liyao Xiang*
Shanghai Jiao Tong University
Shanghai, China
{yinhang_SJTU, xiangliyao08}@sjtu.edu.cn

Dong Ding
Shanghai Jiao Tong University
Shanghai, China
18916162516@sjtu.edu.cn

Yuheng He, Yihan Wu, Pengzhi Chu, Xinbing Wang
Shanghai Jiao Tong University
Shanghai, China
{heyuheng, caracalla, pzchu, xwang8}@sjtu.edu.cn

Chenghu Zhou
Chinese Academy of Sciences
Beijing, China
zhouchsjtu@gmail.com

Abstract

We investigate the entity alignment (EA) problem with unlabeled dangling cases, meaning that partial entities have no counterparts in the other knowledge graph (KG), yet these entities are unlabeled. The problem arises when the source and target graphs are of different scales, and it is much cheaper to label the matchable pairs than the dangling entities. To address this challenge, we propose the framework *Lambda* for dangling detection and entity alignment. Lambda features a GNN-based encoder called KEESA with a spectral contrastive learning loss for EA and a positive-unlabeled learning algorithm called iPULE for dangling detection. Our dangling detection module offers theoretical guarantees of unbiasedness, uniform deviation bounds, and convergence. Experimental results demonstrate that each component contributes to overall performances that are superior to baselines, even when baselines additionally exploit 30% of dangling entities labeled for training.

1 Introduction

Entity alignment is a problem that seeks entities referring to the same real-world identity across different knowledge graphs (KGs), and is widely deployed in fields such as knowledge fusion, question-answering, web mining, etc. To address the issue, embedding-based methods have been proposed to capture entity similarity in the embedding space through translation-based [24, 42, 20] or graph neural network (GNN)-based [41, 37, 27, 26, 35] models. Particularly, if the entities do not have counterparts on another KG, the entities are referred to as *dangling entities*, as shown in Fig. 1.

In many real-world scenarios, the labels for the dangling entities on KGs are often missing, as those labels are much harder to acquire. For example, in KG plagiarism detection, it is relatively easy to align entity pairs that both exist in KGs, but one would have to traverse all possible pairs to conclude an entity is not paired. Hence *EA with unlabeled dangling entities* is a hard but realistic problem. The problem even worsens in EA on KGs of different scales where the dangling entities take a large proportion of all nodes.

Despite that many works have been investigating the EA problem with dangling entities, few have focused on EA with unlabeled dangling cases. We list closely related works in Table 1. The work of [33] extends the conventional EA methods MTransE [5] and AliNet [37] to the case with dangling

*Corresponding author: Liyao Xiang, xiangliyao08@sjtu.edu.cn.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2507

entities, and thus require a portion of labeled dangling entities for training. Based on their works, MHP [19] has improved performance with additional knowledge, i.e., the high-order proximities information for alignment. Both UED [24] and SoTead [23] are unsupervised schemes that rely on side information such as entity name/attribute as global alignment information. Different from prior works, we consider a stricter case where neither side information nor any labeled dangling entities are known, as side information often leads to name-bias [21, 44] while labels for dangling entities are hard to obtain.

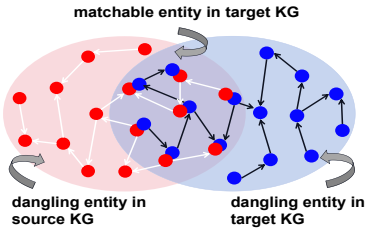

Figure 1: Examples of dangling entities.

Method	Side Info	Dangling Labels
[33] w/ MTransE	✗	✔
[33] w/ AliNet	✗	✔
UED [24]	✔	✗
SoTead [23]	✔	✗
MHP [19]	✗	✔ + high-order info
Our Work	✗	✗

Table 1: Different EA models with dangling cases.

The EA with unlabeled dangling entities faces unique challenges: *first,* the unlabeled dangling entities would cause erroneous information to propagate through neighborhood aggregation if applying conventional GNN-based embedding methods, negatively affecting the dangling detection and alignment of matchable entities. *Second,* the absence of labeled dangling entities makes its feature distribution non-observable, requiring the model to distinguish potential dangling entities while learning the representation of matchable entities. Hence the EA problem has to be solved with mere positive (matchable entities with labels) and unlabeled samples, yet without any prior knowledge of the distribution of the nodes.

We tackle the first challenge by proposing a novel GNN-based EA framework. To eliminate the 'pollution' of dangling entities, the adaptive dangling indicator has been applied globally for selective aggregation. Relation projection attention is designed to combine both entity and relation information for a more comprehensive representation. The designed spectral contrastive learning loss disentangles the matchable entities from dangling ones while portraying a unified embedding space for entity alignment.

As to the second challenge, we first derive an unbiased risk estimator and a tighter uniform deviation bound for the positive-unlabeled (PU) learning loss. However, such an estimator still requires prior knowledge of the proportion of positive entities among all nodes. Thus we develop an iterative strategy to estimate such prior knowledge while training the classifier with a PU learning loss. The prior estimation could also be seen as dangling entity detection; if too few entities are determined to be matchable, one can stop all subsequent procedures and decide the two KGs cannot be aligned.

Our framework Lambda is provided in Fig. 2 where there are two phases corresponding to two trained models — dangling detection and entity alignment. Both phases share one GNN-based encoder and the spectral contrastive learning loss. The dangling detection additionally uses a positive-unlabeled learning loss. The GNN-based encoder contains both the intra-graph and the cross-graph representation learning modules. After the dangling detection, the estimated proportion of matchable entities is figured for judging whether two KGs could be aligned. Only aligned KGs are sent for EA model training, and then only first-phase predicted matchable entity embeddings are obtained from the EA model for inference. Finally, we select pairs of entities that are mutually nearest by their embeddings as aligned pairs.

Highlights of our contributions are as follows: we raise the challenging problem of EA with unlabeled dangling entities for the first time. To resolve the issue, we propose the framework Lambda featured by a GNN-based encoder called KEESA with spectral contrastive learning for EA and a positive-unlabeled learning algorithm for dangling detection called iPULE. We provide a theoretical analysis of PU learning on the unbiasedness, uniform deviation bound, and convergence. Experiments on a variety of real-world datasets have demonstrated our alignment performance is superior to baselines, even the baselines with 30% labeled dangling entities. Our code is available on github[2].

[2]https://github.com/Handon112358/NeurIPS_2024_Learning-Matchable-Prior-For-Entity-Alignment-with-Unlabeled-Dangling-Cases

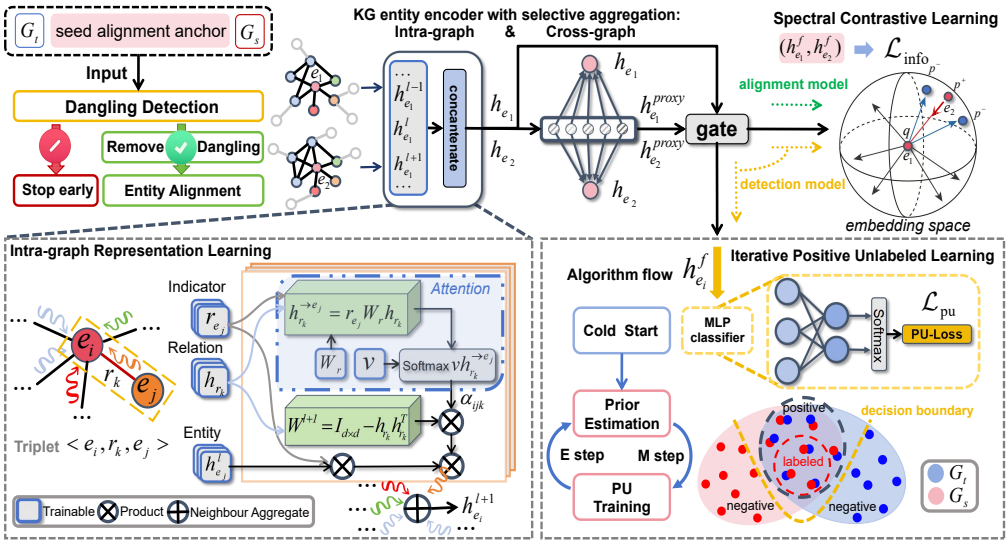

Figure 2: The illustration of our framework.

2 Preliminaries and Related Work

2.1 Entity Alignment

Embedding-based entity alignment methods have evolved rapidly and are gradually becoming the mainstream approach of EA in recent years due to their flexibility and effectiveness [17], which aim to encode KGs into low-dimensional embedding space to capture the similarities of entities [18, 10]. It could be divided into translation-based [24, 42, 20] and GNN-based [41, 37, 27, 26, 35].

Previous EA methods mostly assume a one-to-one correspondence between two KGs, but such an assumption does not always hold and thus leads to a performance drop in real-world cases [38]. Notably, Sun *et al.* [33] as a pioneering work modeled it upon a supervised setting, i.e., a small set of aligned entities and labeled dangling entities. On this basis, MHP [19] employed more dangling information concerning high-order proximities in both training and inference. UED [24] and SoTead [22] propose an unsupervised translation-based method for joint entity alignment and dangling entity detection without labeled dangling entities, while the practical problem of matching cost is ignored.

2.2 Positive-Unlabled Learning

Let $X \in \mathbb{R}^d, d \in \mathbb{N}$, and $Y \in \{\pm 1\}$ be the input and output random variables. We also define $p(x, y)$ to be the *joint probability density of* (X, Y), $p_{\mathrm{p}}(x) = p(x \mid y = +1)$, $p_{\mathrm{n}}(x) = p(x \mid y = -1)$ to be the *P (Positive) and N (Negative) marginals (a.k.a., class-conditional densities)*, and $p_{\mathrm{u}}(x)$ be the *U (Unlabeled) marginal*. The *class-prior probability* is expressed as $\pi_p = p(y = +1)$, which is assumed to be known throughout the paper, and can be estimated from known datasets [13].

The PU learning problem setting is as follows: the positive and unlabeled data are sampled independently from $p_{\mathrm{p}}(x)$ and $p_{\mathrm{u}}(x)$ as $\mathcal{X}_{\mathrm{p}} = \{x_i^{\mathrm{p}}\}_{i=1}^{n_{\mathrm{p}}} \sim p_{\mathrm{p}}(x)$ and $\mathcal{X}_{\mathrm{u}} = \{x_i^{\mathrm{u}}\}_{i=1}^{n_{\mathrm{u}}} \sim p_{\mathrm{u}}(x)$, and a classifier is trained from $\mathcal{X}_{\mathrm{p}}$ and $\mathcal{X}_{\mathrm{u}}$, in contrast to learning a classifier telling negative samples apart from positive ones. The general assumption of the previous work is to let the unlabeled distribution be equal to the overall data distribution, i.e., $p_{\mathrm{u}}(x) = p(x)$ since $p_{\mathrm{u}}(x)$ cannot be obtained, but the assumption hardly holds in many real-world scenarios, for example transductive learning, making methods in [13, 30] infeasible.

3 Selective Aggregation with Spectral Contrastive Learning

Notation: Source and target KG $G_s = (E_s, R_s, T_s)$, $G_t = (E_t, R_t, T_t)$ stored in triples <entity, relation, entity>: entities E, relations R, and triples $T \subseteq E \times R \times E$, $E_s = D_s \cup M_s$, $E_t = D_t \cup M_t$,

https://doi.org/10.52202/079017-2507

where D denotes dangling and M denotes matchable. A set of pre-aligned anchor node pairs are $S = \{(u, v)|u \in M_s, v \in M_t, u \equiv v\}$. (see appendix A for more details).

We start by introducing the KEESA (**K**G **E**ntity **E**ncoder with **S**elective **A**ggregation).

3.1 KG Entity Encoder with Selective Aggregation

Adaptive Dangling Indicator & Relation Projection Attention. Real-world EA tasks often involve graphs with dangling distortion [39, 3]. Conventional GNN aggregation will 'pollute' matchable entities' embeddings with dangling. However, a hard dangling indicator for the entity is over-confident as only approximate results can be obtained without labels. Incorrect indicators may lead to inappropriate aggregation and thus destruction of the KG structure. Instead, we apply a learnable scalar weight r_{e_j} for each e_i's neighboring message:

$$\boldsymbol{h}_{e_i}^{l+1} = \sigma\left(\sum_{e_j \in \mathcal{N}_{e_i} \cup \{e_i\}} \underbrace{\tanh(r_{e_j})}_{\text{adaptive dangling indicator}} \alpha_{i,j} W^{l+1} \boldsymbol{h}_{e_j}^{l}\right), \tag{1}$$

where $\tanh$ serves to normalize r_{e_j} to the range of $[-1, 1]$. The initialization of r_{e_j} is critical, please see the implementation details for more.

As compressed feature of e_j — r_{e_j} is a plain scalar, we link relation r_k's embedding $\boldsymbol{h}_{r_k}$ to the associated entity e_j by $\boldsymbol{h}_{r_k}^{\rightarrow e_j}$ for capturing more comprehensive attention. A matrix $W_r \in \mathbb{R}^{d \times d}$ with an orthogonal regularizer L_o is applied to $\boldsymbol{h}_{r_k}$ to perform projection while preserving its norm for better convergence:

$$\boldsymbol{h}_{r_k}^{\rightarrow e_j} = r_{e_j} W_r \boldsymbol{h}_{r_k} \quad \text{and} \quad L_o = \left\| W_r^\top W_r - I_{d \times d} \right\|_2^2.$$

The attention coefficient is obtained by the following equation, where $\boldsymbol{v}^\top$ is the attention vector:

$$\alpha_{ijk}^{l} = \frac{\exp(\boldsymbol{v}^\top \boldsymbol{h}_{r_k}^{\rightarrow e_j})}{\sum_{e_m \in \mathcal{N}_{e_i}, <e_i, r_n, e_m> \in T_s \cup T_t} \exp(\boldsymbol{v}^\top \boldsymbol{h}_{r_n}^{\rightarrow e_m})} \tag{2}$$

Intra- & Cross-Graph Representation Learning. Based on the above, we can express the embedding of e_i at the $(l+1)$-th layer $\boldsymbol{h}_{e_i}^{l+1}$ as Eq. 1, where W^{l+1} is specified as $W^{l+1} = I_{d \times d} - 2\boldsymbol{h}_{r_k} \boldsymbol{h}_{r_k}^\top$ by the triplet $< e_i, r_k, e_j >$ inclusive relation embedding $\boldsymbol{h}_{r_k}$. We adopt the $\tanh(\cdot)$ as the activation function. The Householder transformation W^{l+1} is applied on the last layer embedding $\boldsymbol{h}_{e_i}^{l}$ to restore the useful relative positions of KG entities at each layer recursively.

Overall, the *intra-graph representation* $\boldsymbol{h}_{e_i}$ of e_i is obtained by concatenating embeddings from all layers. Its *cross-graph representation* $\boldsymbol{h}_{e_i}^{proxy}$ can be described by $\boldsymbol{h}_{e_i}$ and proxy $\boldsymbol{q}_j$, where the latter is generated by *Proxy Matching Attention Layer* [25] to align the embeddings across two graphs. With S_p representing the set of proxy vectors, and $\text{sim}(\cdot)$ denoting the cosine similarity, we have:

$$\boldsymbol{h}_{e_i} = [\boldsymbol{h}_{e_i}^0 \| \boldsymbol{h}_{e_i}^1 \| ... \| \boldsymbol{h}_{e_i}^l] \quad \text{and} \quad \boldsymbol{h}_{e_i}^{proxy} = \sum_{\boldsymbol{q}_j \in S_p} \frac{\exp(\text{sim}(\boldsymbol{h}_{e_i}, \boldsymbol{q}_j))}{\sum_{\boldsymbol{q}_k \in S_p} \exp(\text{sim}(\boldsymbol{h}_{e_i}, \boldsymbol{q}_k))} (\boldsymbol{h}_{e_i} - \boldsymbol{q}_j).$$

Finally, we employ a gating mechanism [31] to integrate both intra-graph representation $\boldsymbol{h}_{e_i}$ and cross-graph representation $\boldsymbol{h}_{e_i}^{proxy}$ into $\boldsymbol{h}_{e_i}^{f}$:

$$\boldsymbol{\theta}_{e_i} = \text{sigmoid}(\boldsymbol{W}_g \boldsymbol{h}_{e_i}^{proxy} + \boldsymbol{b}), \qquad \boldsymbol{h}_{e_i}^{f} = [(\boldsymbol{\theta}_{e_i} \cdot \boldsymbol{h}_{e_i} + (1 - \boldsymbol{\theta}_{e_i}) \cdot \boldsymbol{h}_{e_i}^{proxy}) \| r_{e_i}],$$

where $\boldsymbol{W}_g$ and $\boldsymbol{b}$ are the gate weight and gate bias, respectively. The learnable weight of e_i is also attached to the embedding. It is worth noticing that for each entity on either G_s or G_t, they are encoded by one shared KEESA with below spectral contrastive learning for a unified representation space.

3.2 Spectral Contrastive Learning

In this part, we propose the spectral contrastive learning loss $\mathcal{L}_{\text{info}}$ with high-quality negative sample mining, which serves both tasks (entity alignment and dangling detection) at the same time.

Specifically, given a pre-aligned matchable entity $e_i \in \mathcal{X}_p$, let there be a paired positive sample entity $e_+^i \in \mathcal{X}_p$, such that $(e_i, e_+^i) \in S$, and N sampled entity $\{e_j^i\}^N$ as negative samples $(e_i, e_j^i) \notin S$. The spectral contrastive learning loss is one specific form of alignment loss $H(\cdot)$:

$$\mathcal{L}_{\text{info}} = \sum_{e_i \in \mathcal{X}_p} \log \left[1 + \sum_{j}^{N} \exp(\lambda\, H(e_i, e_+^i, e_j^i)) \right]. \tag{3}$$

Unified Representation for Entity Alignment. We expect a unified feature space where the distance between aligned anchor node pairs is as close as possible, while the unaligned is on the contrary. To satisfy this intuition, we introduce an alignment loss:

$$H(e_i, e_+^i, e_j^i) = [\text{sim}(e_i, e_j^i) - \text{sim}(e_i, e_+^i) + \gamma]_+, \tag{4}$$

where $[x]_+$ represents $\max(0, x)$ and $\text{sim}(\cdot)$ indicates L_2-norm distance between the embeddings. A soft margin γ is involved to discourage trivial solutions, e.g., $\text{sim}(e_i, e_j^i) = \text{sim}(e_i, e_+^i) = 0$.

Discrimination for Dangling Detection. For our proposed dangling detection, the vital task is to discriminate the dangling from the matchable ones with unlabeled dangling entities. Hence unsupervised method of spectral clustering is exploited to separate two types of entities. We design the loss function according to Lemma 1 to achieve its equivalent effect.

Lemma 1. *Given one positive sample p^+ for q, and N negative samples $\{p_j^-\}^N$ (node set: $\{q, p^+\} \cup \{p_j^-\}^N$), employing the following loss function is equivalent to conducting spectral clustering on similarity graph π with the temperature hyper-parameter λ:*

$$\textit{infoNCE}(q, p^+, \{p^-\}^N) = -\log \frac{\exp(\lambda\ \textit{sim}(q, p^+))}{\exp(\lambda\ \textit{sim}(q, p^+)) + \sum_j^N \exp(\lambda\ \textit{sim}(q, p_j^-))}. \tag{5}$$

The equivalence has been discussed in previous studies [40, 1, 6, 32]. Regarding our proposed problem, the positive samples are the corresponding pairs whereas the negative samples are those sampled unaligned pairs. The equivalence is derived as follows:

$$\begin{aligned}
\textit{infoNCE}(q, p^+, \{p^-\}^N) &= \log \frac{\exp(\lambda \text{sim}(q, p^+)) + \sum_j^N \exp(\lambda \text{sim}(q, p_j^-))}{\exp(\lambda\ \text{sim}(q, p^+))} \\
&= \log[1 + \frac{\sum_j^N \exp(\lambda\ \text{sim}(q, p_j^-))}{\exp(\lambda\ \text{sim}(q, p^+))}] \\
&= \log \left[1 + \sum_j^N \exp(\lambda\ \text{sim}(q, p_j^-) - \lambda\ \text{sim}(q, p^+)) \right].
\end{aligned}$$

The spectral contrastive learning loss could be obtained by replacing the exponent term with Eq. 4.

Remark. In the alignment loss function, we observe that dangling entities are only in the negative samples, and the entities in the positive samples are all matchable. Such a stark asymmetry provides an advantage in distinguishing between dangling and matchable entities.

We also prove that $\mathcal{L}_{\text{info}}$ (Eq. 3) can promote model learning by Lemma 2 (see appendix B for proof).

Lemma 2. *The loss $\mathcal{L}_{\text{info}}$ can mine high-quality negative samples, which we show has an equivalent effect to truncated uniform negative sampling (TUNS) in [35].*

Minimizing the spectral contrastive loss of Eq. (3) maps matchable and dangling entities into a unified but distinguishable feature space for improved entity alignment while facilitating dangling detection. In practice, we adopt the loss normalization trick [8] on $H(\cdot)$ to speed up training.

4 Iterative Positive-Unlabeled Learning for Dangling Detection

We expect to avoid any additional computational overhead for EA if few entities are matchable for the source and target KG. Thus, we address a more challenging problem in EA: given partial pre-aligned

 https://doi.org/10.52202/079017-2507

matchable entities as positive samples (i.e., $\mathcal{X}_p$), how to jointly predict the proportion of matchable entities in the unlabeled nodes (i.e., $\pi_{\mathrm{p}}^{\mathrm{u}}$) and identify those entities? If $\pi_{\mathrm{p}}^{\mathrm{u}}$ could be predicted, it could serve as an indicator whether we should proceed to EA. We propose to address the problem by PU learning.

Unbiased Risk Estimator. First, we propose a new unbiased estimation for PU learning without any constraint (i.e., $p_{\mathrm{u}}(x) = p(x)$ in [13, 30]) on unlabeled samples distribution $p_{\mathrm{u}}(x)$ concerning the overall distribution $p(x)$. Assuming that π_{p} and $\pi_{\mathrm{p}}^{\mathrm{u}}$ are known (estimation strategy would be given later), we have:

Theorem 1. *Suppose that $g \in \mathcal{G} : \mathbb{R}^d \to \mathbb{R}$ is a binary classifier, and $\ell : \mathbb{R} \times \{\pm 1\} \to \mathbb{R}$ is the loss function by which $\ell(t, y)$ means the loss incurred by predicting an output t when the ground truth is y. $\widehat{R}_{\mathrm{pu}}(g)$ is the **unbiased risk estimator** of $R(g)$:*

$$\widehat{R}_{\mathrm{pu}}(g) = \pi_{\mathrm{p}}\widehat{R}_{\mathrm{p}}^{+}(g) + \frac{\pi_{\mathrm{n}}}{\pi_{\mathrm{n}}^{\mathrm{u}}} \cdot \left[\widehat{R}_{\mathrm{u}}^{-}(g) - \pi_{\mathrm{p}}^{\mathrm{u}}\widehat{R}_{\mathrm{p}}^{-}(g)\right], \tag{6}$$

where $\pi_{\mathrm{n}} = p(y = -1)$ and $\pi_{\mathrm{n}}^{\mathrm{u}} = p_{\mathrm{u}}(y = -1)$ are estimable class priors given π_{p} and $\pi_{\mathrm{p}}^{\mathrm{u}}$, $R_p^+(g) = \mathbb{E}_{X\sim p_{\mathrm{p}}(x)}[\ell(g(X), +1)]$ and $R_n^-(g) = \mathbb{E}_{X\sim p_{\mathrm{n}}(x)}[\ell(g(X), -1)]$ (see appendix C for proof).

By our proof, $\widehat{R}_{\mathrm{pu}}(g)$ is an unbiased risk bound for the PU learning. More importantly, the bound provided by Thm. 2 is a tighter uniform deviation bound than the classic *Non-negative Risk Estimator* [13]:

Theorem 2. *Let $\mathrm{Var}(R)$ denote the uniform deviation bound of risk estimator R, and Non-negative Risk Estimator be $\widehat{R'}_{\mathrm{pu}}(g)$, then:$\mathrm{Var}(\widehat{R}_{\mathrm{pu}}(g)) < \mathrm{Var}(\widehat{R'}_{\mathrm{pu}}(g))$(see appendix D for proof).*

Positive Unlabeled Loss Function. Since it is evident that all negative samples exist in unlabeled data, we have $\frac{\pi_{\mathrm{n}}}{\pi_{\mathrm{n}}^{\mathrm{u}}} < 1$ and thus we apply a hyper-parameter $\alpha = \frac{\pi_{\mathrm{n}}^{\mathrm{u}}}{\pi_{\mathrm{n}}}$ to scale $\pi_{\mathrm{p}}\widehat{R}_{\mathrm{p}}^{+}(g)$ equivalently and $\max(\cdot)$ to restrict the estimated $\pi_{\mathrm{n}}R_{\mathrm{n}}^{-}(g) \geq 0$. The PU learning loss function is formulated as:

$$\mathcal{L}_{\mathrm{pu}} = \alpha\pi_{\mathrm{p}}\widehat{R}_{\mathrm{p}}^{+}(g) + \max\{0, \widehat{R}_{\mathrm{u}}^{-}(g) - \pi_{\mathrm{p}}^{\mathrm{u}}\widehat{R}_{\mathrm{p}}^{-}(g)\},$$

We specify the corresponding risk function using cross-entropy losses as below:

$$\widehat{R}_{\mathrm{p}}^{+}(g) = \frac{1}{|\mathcal{X}_p|}\sum_{e_i\in\mathcal{X}_p} \log \hat{y}_i(+1), \widehat{R}_{\mathrm{u}}^{-}(g) = \frac{1}{|\mathcal{X}_u|}\sum_{e_i\in\mathcal{X}_u} \log \hat{y}_i(-1), \widehat{R}_{\mathrm{p}}^{-}(g) = \frac{1}{|\mathcal{X}_p|}\sum_{e_i\in\mathcal{X}_p} \log \hat{y}_i(-1)$$

where the output logit for $e_i \in E_s \cup E_t$, being labeled as a state $u \in \{+1, -1\}$, is $\hat{y}_i(u) = softmax(\mathbf{MLP}(h_{e_i}^f))$, based on KEESA output $h_{e_i}^f$. Hence each term in the final loss can be calculated or estimated without the negative labels.

Iterative PU Learning with Prior Estimator How could we estimate prior π_{p} and $\pi_{\mathrm{p}}^{\mathrm{u}}$? Inspired by [11], we introduce a hidden variable in the model as well as an iterative approach. We adopt a variational approximation strategy and a warm-up phase to tackle the cold start problem, as shown in Alg. 1. First, we estimate and fix the class prior π_{p} and $\pi_{\mathrm{p}}^{\mathrm{u}}$ by the ratio of the anchor points in the training set. $\mathcal{L}_{\mathrm{info}}$ is optimized together with $\mathcal{L}_{\mathrm{pu}}$ for a discriminative embedding space in the warm-up phase. Finally, we minimize $\mathcal{L}_{\mathrm{pu}}$ to update the class prior and the model parameters alternately till convergence.

The convergence guarantee is provided in Thm. 3, which mostly follows the convergence of EM algorithm. We collect the proof in appendix E.

Theorem 3. *Given the assumptions of marginalization in Eq. 16 and Eq. 17, the objective function of $-\mathcal{L}_{\mathrm{pu}}$ is the same as the expectation function Q of Eq. 13 where the loss function is the cross entropy function $CE(\bar{y}_i, \hat{y}_i) = -\bar{y}_i(+1)\log\hat{y}_i(+1) - \bar{y}_i(-1)\log\hat{y}_i(-1)$ on the **preference condition**: $\sum_{j\in\mathcal{U}}\frac{1}{|\mathcal{U}|}\log\frac{\hat{y}_j(+1)}{\hat{y}_j(-1)} \approx \sum_{i\in\mathcal{P}}\frac{1}{|\mathcal{P}|}\log\frac{\hat{y}_i(+1)}{\hat{y}_i(-1)}$.*

The iterative process of our method is a special case of the EM algorithm. We hold the same assumptions as the EM algorithm and we further assume the training of f is able to find the globally optimal θ. Although the assumptions seem to be too strict, the algorithm typically converges in practice as we verified in the experimental section.

Algorithm 1 iPULE (**i**terative **PU** **L**earning with Prior **E**stimator)

Require: G_s and G_t are treated as one input graph $G = (\mathcal{V}, \mathcal{E})$, positive-node set $\mathcal{P} = \mathcal{X}_p$, unlabeled-node set $\mathcal{U} = \mathcal{X}_u$, classifier f with initial parameters θ_0, KEESA Enc(G, ψ) with initial parameters ψ_0 and warm-up epoch N. $\mathcal{L}$ represents training loss.

Ensure: Model parameters θ, ψ and estimated prior $\hat{\pi}_{\mathrm{p}}$ and $\hat{\pi}_{\mathrm{p}}^{\mathrm{u}}$

$l \leftarrow \infty, \quad \hat{\pi}_{\mathrm{p}}^{\mathrm{u}} \leftarrow \hat{\pi}_{\mathrm{p}} \leftarrow \frac{|\mathcal{P}|}{|\mathcal{P}|+|\mathcal{U}|}, \quad i \leftarrow 0, \quad \beta = \beta_0;$ //Initial value

$\mathcal{L} \leftarrow \beta \cdot \mathcal{L}_{\mathrm{info}} + (1-\beta) \cdot \mathcal{L}_{\mathrm{pu}};$ //Loss of warm-up

repeat

  $\mathbf{X} \leftarrow \mathrm{Enc}(G, \psi);$ //Entity embedding matrix $\mathbf{X}$

  $\theta, \psi \leftarrow \arg\min_{\theta,\psi} \mathcal{L}(\theta; \mathbf{X}, \mathbf{y}, \mathcal{P}, \mathcal{U});$ //Optimize Enc$(\cdot)$ and f jointly

  $l \leftarrow \mathcal{L}(\theta; \mathbf{X}, \mathbf{y}, \mathcal{P}, \mathcal{U});$

until N epochs is over //Warm-up phase to solve cold start

$\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{pu}};$

repeat

  $\mathbf{X} \leftarrow \mathrm{Enc}(G, \psi), \quad \hat{y}_i \leftarrow f(\mathbf{X}, i; \theta) \text{ for all } i \in \mathcal{V};$

  $\hat{\pi}_{\mathrm{p}}^{\mathrm{u}} \leftarrow |\mathcal{U}|^{-1} \sum_{i \in \mathcal{U}} \mathbb{I}[\hat{y}_i(+1) > 0.5], \quad \hat{\pi}_{\mathrm{p}} \leftarrow \frac{|\mathcal{P}|+|\mathcal{U}| \cdot \hat{\pi}_{\mathrm{p}}^{\mathrm{u}}}{|\mathcal{P}|+|\mathcal{U}|};$ //E step

  $l' \leftarrow l, \quad l \leftarrow \mathcal{L}(\theta; \mathbf{X}, \mathbf{y}, \mathcal{P}, \mathcal{U});$

  $\theta, \psi \leftarrow \arg\max_{\theta,\psi} -\mathcal{L}(\theta; \mathbf{X}, \mathbf{y}, \mathcal{P}, \mathcal{U});$ //M step

until $|l' - l|$ converge **OR** $\hat{\pi}_{\mathrm{p}}$ converge

return

5 Experiments

Our method is evaluated on datasets GA16K, DBP2.0, and GA-DBP15K. DBP2.0 and GA-DBP15K are used for the verification of iPULE. To address incomparability caused by inconsistent metrics, we adopt the GA16K dataset to enable compromised comparison of the Dangling-Entities-Unaware EA method. We further compare our method with dangling aware baselines on DBP2.0. *Statistics of experimental dataset* in appendix F, and *additional experiment* in appendix G.

Datasets. The training/test sets for each dataset are generated using a fixed random seed. For entity alignment, 30% of matchable entity pairs constitute the training set, while the remaining form the test set. For dangling entity detection, we did not utilize any labeled dangling entity data, in contrast to prior work which labels an extra 30% of the dangling entities for training [33, 19].

Baselines. Since our work does not take advantage of any side information, we emphasize its comparison with the previous methods purely depending on graph structures. These works majorly incorporate two types:

Dangling-Entities-Unaware. We include advanced entity alignment methods in recent years: GCN-Align [41], RSNs [9], MuGNN [4], KECG [16]. Methods with bootstrapping to generate semi-supervised structure data are also adopted: BootEA [35], TransEdge [36], MRAEA [26], AliNet [37], and Dual-AMN [25].

Dangling-Entities-Aware. To the best of our knowledge, the method of [33] is the most fairly comparable baseline which is based on MTransE [5] and AliNet [37]. Because MHP [19] over-emphasized more use of labeled dangling data like high-order similarity information which is also based on the above two methods, while SoTead [22] and UED [24] utilize additional side-information. SoTead [22] and UED [24] can only execute the degraded version on DBP2.0 cause no side-information is available on that. We exclude them from baselines for our methods. [33] introduces three techniques to address the dangling entity issue: nearest neighbor (NN) classification, marginal ranking (MR), and background ranking (BR).

Implementation Detail. We use the `Keras` framework for developing our approach. Our experiments are conducted on a workstation with an NVIDIA Tesla A100 GPU, and 80GB memory.

By default, the embedding dimension is set to 128 with the depth of GNN set to 2 and a dropout rate of 0.3. A total of 64 proxy vectors are used and margin γ = 1. RMSprop optimizer is adopted with a learning rate of 0.005 and batch size of 5120. λ is set to be 30. β is set as 1e-3 for all datasets. CSLS [14] is adopted as the distance metric for alignment. As we found, the $\tanh$ function changes rapidly

https://doi.org/10.52202/079017-2507

in the region close to 0 but stays stable in the region beyond $[-3, 3]$. Hence we initialize the r_{e_j} to 1 to prevent gradients oscillation or near-zero gradients.

5.1 Experiments of iPULE Convergence and Class Prior Estimation

DBP2.0 π_{p} between 20%-50% contains more entities to be aligned, trained by pre-aligned 9%-15% nodes then judged by iPULE as aligned KGs. GA-DBP15K π_{p} between 10%-25% are treated as unaligned KGs ignoring the pre-aligned part, trained by all pre-aligned 10%-25% nodes. We get accurate estimation and convergence results as shown in Fig. 3. As iPULE progresses, the estimated class prior gradually approaches the true value as the first row for GA-DBP15K and the second for DBP2.0. $\pi_{\mathrm{p}}^{\mathrm{u}} = 0$ for GA-DBP15K while DBP2.0's are given by red dotted line respectively. The π_{p} for GA-DBP15K is stably consistent as pre-aligned proportion due to accurate estimation of its $\pi_{\mathrm{p}}^{\mathrm{u}}$. As common in PU learning [43], iPULE treats more nodes as positive when $\pi_{\mathrm{p}} \approx 50\%$ in FR-EN.

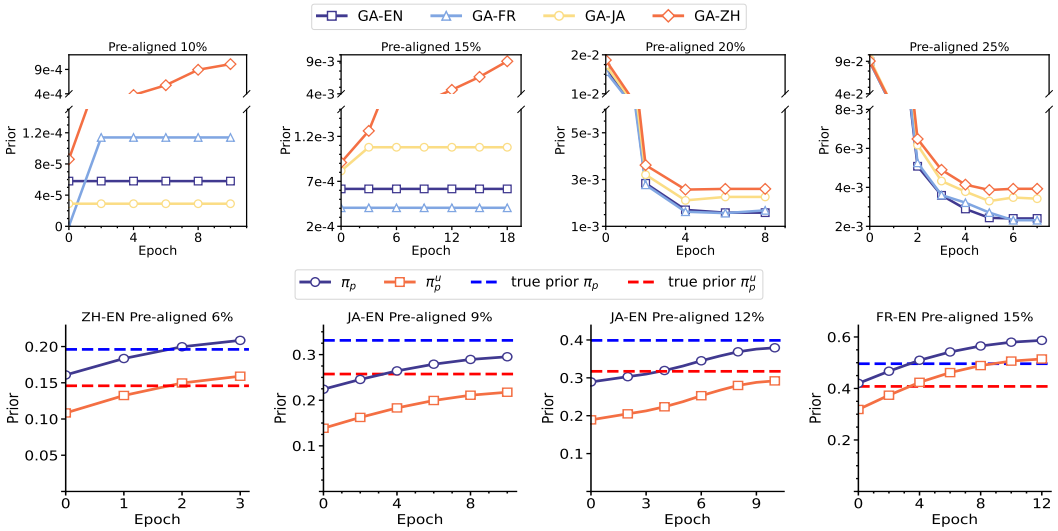

Figure 3: Prior estimation GA-DBP15K and DBP2.0. (loss convergence in appendix F).

5.2 Experiments Unaware of Dangling Entities

We show the experiments on baselines without considering dangling entities in this section.

Dangling-Entities-Unaware Baselines Comparison. The direct comparison between our method and the dangling-entities-unaware baselines is unavailable due to inconsistent metrics used. Hence, we adopt the GA16K dataset as a compromise and do not remove any detected dangling entities for entity alignment. Thus the ranking list S in Hits@K only contains (matchable) entities in the source graph since GA16K only contains dangling entities in the target KG. In Tab. 2, Dual-AMN demonstrates a competitive performance but is inferior to ours at Hits@1. MRAEA performs similarly to Dual-AMN since the latter is built on the former. TransEdge performs poorly since the method adopts semi-supervised bootstrapping to mine anchor entities iteratively. The presence of dangling entities could lead to false anchors and spread of error. Meanwhile, it is also a relation-centric approach that suffers from insufficient relation information on GA16K. Other baselines exhibit up-to-par performance but our method delivers consistently superior or state-of-the-art Hits@Ks.

5.3 Experiments Aware of Dangling Entities

We provide a comparison of dangling detection and entity alignment with baselines aware of dangling.

Dangling Entities Detection Performance. We test our method's dangling detection performance compared with baselines aware of dangling entities. The results on DBP2.0 in the consolidated setting are reported in Tab. 12. Note that the comparison is unfair as we don't use 30% of the labeled

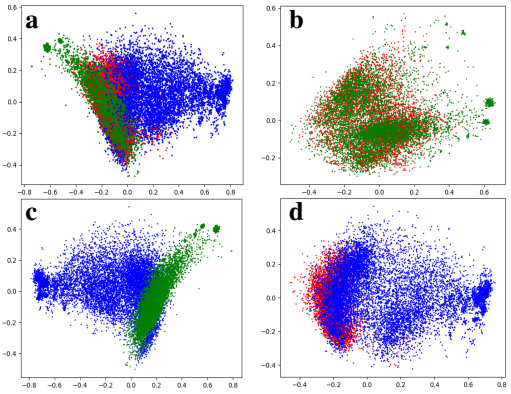

Figure 4: Visualization of entity representations learned by our method on GA16K dataset.

Method	GA16K		
	H@1	H@10	H@50
BootEA	13.95	37.25	49.08
TransEdge	0.03	0.12	0.14
MRAEA	63.97	76.64	81.06
GCN-Align	29.48	45.64	57.15
RSNs	9.40	42.70	46.70
MuGNN	62.17	76.25	80.87
KECG	44.18	57.73	63.41
AliNet	48.53	67.72	74.50
Dual-AMN	64.49	**80.55**	**84.67**
Ours	**67.59**	80.33	84.35

Table 2: Performance comparison with dangling-entities-unaware baselines on GA16K.

Methods		ZH-EN			EN-ZH			JA-EN			EN-JA			FR-EN			EN-FR		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
AliNet	NNC	.676	.419	.517	.738	.558	.634	.597	.482	.534	.761	.120	.207	.466	.365	.409	.545	.162	.250
	MR	.752	.538	.627	.828	.505	.627	.779	.580	.665	.854	.543	.664	.552	.570	.561	.686	.549	.609
	BR	.762	.556	.643	.829	.515	.635	.783	.591	.673	.846	.546	.663	.547	.556	.552	.674	.556	.609
MTransE	NNC	.604	.485	.538	.719	.511	.598	.622	.491	.549	.686	.506	.583	.459	.447	.453	.557	.543	.550
	MR	.781	.702	.740	.866	.675	.759	.799	.708	.751	.864	.653	.744	.482	.575	.524	.639	.613	.625
	BR	**.811**	.728	.767	**.892**	.700	.785	**.816**	.733	.772	**.888**	.731	.801	.539	.686	.604	.692	.735	.713
Ours		.763	**.925**	**.836**	.844	**.909**	**.875**	.807	**.836**	**.821**	.880	**.809**	**.843**	**.615**	**.772**	**.685**	**.732**	**.749**	**.740**

Table 3: Dangling detection results on DBP2.0 in the consolidated setting.

Methods		ZH-EN			EN-ZH			JA-EN			EN-JA			FR-EN			EN-FR		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
AliNet	NNC	.121	.193	.149	.085	.138	.105	.113	.146	.127	.067	.208	.101	.126	.148	.136	.086	.161	.112
	MR	.207	.299	.245	.159	.320	.213	.231	.321	.269	.178	.340	.234	.195	.190	.193	.160	.200	.178
	BR	.203	.286	.238	.155	.308	.207	.223	.306	.258	.170	.321	.222	.183	.181	.182	.164	.200	.180
MTransE	NNC	.164	.215	.186	.118	.207	.150	.180	.238	.205	.101	.167	.125	.185	.189	.187	.135	.140	.138
	MR	.302	.349	.324	.231	.362	.282	.313	.367	.338	.227	.366	.280	.260	.220	.238	.213	.224	.218
	BR	**.312**	.362	.335	**.241**	.376	.294	.314	.363	.336	**.251**	.358	.295	**.265**	.208	.233	**.231**	.213	.222
Ours		.279	**.447**	**.344**	.219	**.489**	**.303**	**.324**	**.409**	**.362**	.234	**.460**	**.310**	.234	**.320**	**.271**	.192	**.363**	**.251**

Table 4: Entity alignment results on DBP2.0 in the consolidated setting.

dangling entities as the baselines. Nevertheless, our approach maintains SOTA performance across all six datasets, excelling in almost every metric except for a slightly inferior precision.

Dangling-Entities-Aware Baselines Comparison. Tab. 4 reports the entity alignment performance comparison in the consolidated setting on DBP2.0. The precision, recall, and F1 scores are computed according to Eq. (19), (20), (21) in **Metric** part of appendix F, respectively. We test the entity alignment performance of our method in comparison with baselines that are aware of dangling entities. Our method still maintains almost state-of-the-art performance, but there is still a slightly inferior precision problem. It makes us wonder about the reasons behind it.

How does our method work? To understand why our method works and its precision slightly suffers, we visualized all entity embeddings of GA16K in Fig. 4. As shown above, matchable entities are denoted as red and green in source and target KG respectively, while dangling as blue. The distribution of three types of entity in Fig. 4(a) suggests our method maps all nodes into a unified embedding space where matchable entities exhibit considerable overlap and are appropriately aligned (shown in Fig. 4(b)). Fig. 4(c)(d) depicts that a part of the dangling entities is intertwined with the matchable ones, suggesting that this part resides at the decision boundary and easily leads to false positives which explain the lesser precision of our method.

https://doi.org/10.52202/079017-2507

5.4 Ablation Studies and Varying Anchor Nodes

We conduct ablation studies to show each module's impact, and similarly pre-aligned entities' impact.

Ablation Studies The impact of adaptive dangling indicator and relation projection attention in our method are investigated. We denote the counterpart removing r_{e_i} as $w/o\ r_{e_i}$, and replacing $\boldsymbol{h}_{r_k}^{\rightarrow e_j}$ with the original $\boldsymbol{h}_{r_k}$ as $w/o\ \boldsymbol{h}_{r_k}^{\rightarrow e_j}$. Fig. 5 gives the ablation study results on DBP2.0, where 'Ours' represents an all-inclusive model. We observe that the $\boldsymbol{h}_{r_k}^{\rightarrow e_j}$ has a more substantial impact than r_{e_i} to the alignment performance. As to why the r_{e_i} has a minor impact on the alignment, we consider it may be attributed to the lower degrees of dangling entities on DBP2.0. The degrees of dangling entities are generally lower than that of matchable ones, indicating that the dangling is more isolated in the graph and thus has less impact on matchable nodes in the neighborhood aggregation.

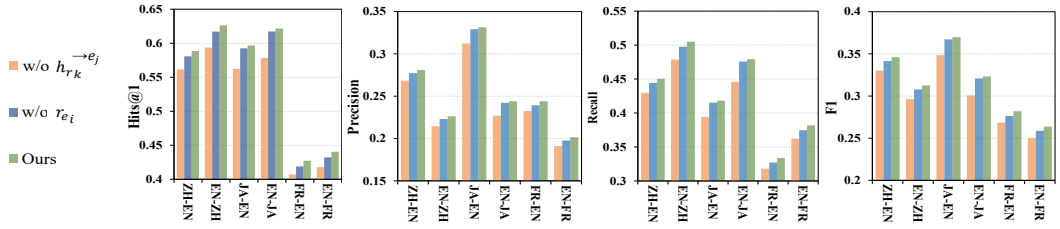

Figure 5: The ablation study of entity alignment performance in the consolidated setting on DBP2.0.

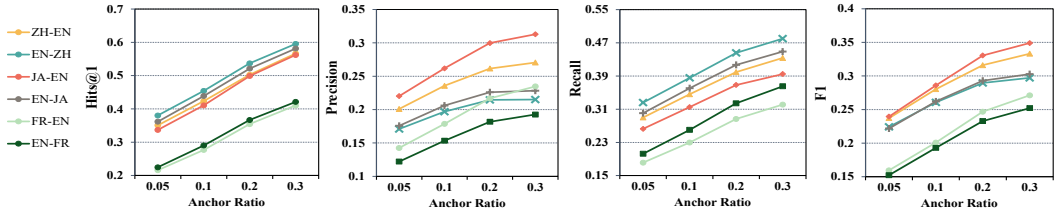

Figure 6: The entity alignment performance on varying pre-aligned anchor nodes ratios on DBP2.0.

Varying Anchor Nodes. Pre-aligned entities may be far scarce in reality. The sensitivity of our method to the proportion variation of anchor nodes is investigated. As the proportion increases, the alignment performance enhances as provided in Fig. 6.

Notably, even with an anchor ratio as low as 5%, our alignment accuracy still well exceeds 30% on most datasets except for FR-EN and EN-FR. Cause they contain twice as many entities and triples as ZH-EN and JA-EN, which introduces intricate dependencies among entities and thus greater challenges in alignment. Moreover, a larger graph may require a higher dimension of representations to learn, but the embedding dimension is restricted to merely 96 due to the out-of-memory problem.

6 Conclusion

We found that previous EA methods suffer from great performance decline if dangling entities are considered. Our goal is to address the EA problem with unlabeled dangling entities. A novel framework Lambda for detecting dangling entities and then pairing alignment is proposed. The core idea is to perform selective aggregation with spectral contrastive learning and to adopt theoretically guaranteed PU learning to relieve the dependence on the labeled dangling entities. Experimental results on multiple representative datasets demonstrate the effectiveness of our proposed approach. This work also has important implications for real-world applications, such as EA of different scales, KG plagiarism detection, etc.

https://doi.org/10.52202/079017-2507

Acknowledgments and Disclosure of Funding

The research was supported in part by NSF China (No. 61960206002, 62272306, 62032020, 62136006).

The authors would like to thank the reviewers for their constructive comments and appreciate the Student Innovation Center of SJTU for providing GPUs. Hang Yin personally thanks Chenyu Liu, Yuting Feng, Jingyuan Zhou, and Qingyang Liu for feedbacks on early versions of this paper. Hang Yin would also like to thank Professor Yuan Luo for his inspiration in the information theory course (CS7317) of Shanghai Jiao Tong University.

References

[1] Hugues Van Assel, Thibault Espinasse, Julien Chiquet, and Franck Picard. A probabilistic graph coupling view of dimension reduction. In *Adcances in neural information processing systems (NeurIPS)*, pages 10696–10708, 2022.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *international semantic web conference (ISWC)*, pages 722–735, 2007.

[3] Khalid Belhajjame and Mohamed-Yassine Mejri. Online maintenance of evolving knowledge graphs with rdfs-based saturation and why-provenance support. *Journal of Web Semantics (JoWS)*, 78:100796, 2023.

[4] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Juanzi Li, and Tat-Seng Chua. Multi-channel graph neural network for entity alignment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1452–1461, 2019.

[5] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1511–1517, 2017.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning (ICML)*, pages 1597–1607, 2020.

[7] Cheng Deng, Yuting Jia, Hui Xu, Chong Zhang, Jingyao Tang, Luoyi Fu, Weinan Zhang, Haisong Zhang, Xinbing Wang, and Chenghu Zhou. Gakg: A multimodal geoscience academic knowledge graph. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM)*, pages 4445–4454, 2021.

[8] Yunjun Gao, Xiaoze Liu, Junyang Wu, Tianyi Li, Pengfei Wang, and Lu Chen. Clusterea: Scalable entity alignment with stochastic training and normalized mini-batch similarities. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 421–431, 2022.

[9] Lingbing Guo, Zequn Sun, and Wei Hu. Learning to exploit long-term relational dependencies in knowledge graphs. In *International conference on machine learning (ICML)*, pages 2505–2514, 2019.

[10] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and learning systems (TNNLS)*, 33(2):494–514, 2021.

[11] Yibo Jiang and Bryon Aragam. Learning nonparametric latent causal graphs with unknown interventions. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2016.

 https://doi.org/10.52202/079017-2507

[13] Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1675–1685, 2017.

[14] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations (ICLR)*, 2018.

[15] Kasper Green Larsen and Jelani Nelson. Optimality of the johnson-lindenstrauss lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 633–638, 2017.

[16] Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, pages 2723–2732, 2019.

[17] Yangning Li, Jiaoyan Chen, Yinghui Li, Yuejia Xiang, Xi Chen, and Hai-Tao Zheng. Vision, deduction and alignment: An empirical study on multi-modal knowledge graph alignment. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[18] Yangning Li, Yinghui Li, Xi Chen, Hai-Tao Zheng, and Ying Shen. Active relation discovery: Towards general and label-aware open relation extraction, 2023.

[19] Juncheng Liu, Zequn Sun, Bryan Hooi, Yiwei Wang, Dayiheng Liu, Baosong Yang, Xiaokui Xiao, and Muhao Chen. Dangling-aware entity alignment with mixed high-order proximities. *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022.

[20] Xiao Liu, Haoyun Hong, Xinghao Wang, Zeyi Chen, Evgeny Kharlamov, Yuxiao Dong, and Jie Tang. Selfkg: Self-supervised entity alignment in knowledge graphs. In *ACM Web Conference (WWW)*, pages 860–870, 2022.

[21] Xiaoze Liu, Junyang Wu, Tianyi Li, Lu Chen, and Yunjun Gao. Unsupervised entity alignment for temporal knowledge graphs. In *Proceedings of the ACM Web Conference 2023*, pages 2528–2538, 2023.

[22] Gongxu Luo, Jianxin Li, Hao Peng, Carl Yang, Lichao Sun, Philip S. Yu, and Lifang He. Graph entropy guided node embedding dimension selection for graph neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI*, pages 2767–2774, 2021.

[23] Shengxuan Luo, Pengyu Cheng, and Sheng Yu. Semi-constraint optimal transport for entity alignment with dangling cases, 2022.

[24] Shengxuan Luo and Sheng Yu. An accurate unsupervised method for joint entity alignment and dangling entity detection. In *Findings of the Association for Computational Linguistics (ACL)*, pages 2330–2339, 2022.

[25] Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. Boosting the speed of entity alignment 10×: Dual attention matching network with normalized hard sample mining. In *Proceedings of the Web Conference 2021 (WWW)*, pages 821–832, 2021.

[26] Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. Mraea: an efficient and robust entity alignment approach for cross-lingual knowledge graph. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, pages 420–428, 2020.

[27] Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. Relational reflection entity alignment. In *ACM International Conference on Information & Knowledge Management (CIKM)*, pages 1095–1104, 2020.

[28] Xinnian Mao, Wenting Wang, Yuanbin Wu, and Man Lan. Lightea: A scalable, robust, and interpretable entity alignment framework via three-view label propagation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 825–838, 2022.

[29] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[30] Gang Niu, Marthinus Christoffel Du Plessis, Tomoya Sakai, Yao Ma, and Masashi Sugiyama. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 1199–1207, 2016.

[31] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks, 2015.

[32] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 6398–6407, 2020.

[33] Zequn Sun, Muhao Chen, and Wei Hu. Knowing the no-match: Entity alignment with dangling cases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 3582–3593, 2021.

[34] Zequn Sun, Wei Hu, and Chengkai Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference (ISWC)*, pages 628–644, 2017.

[35] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping entity alignment with knowledge graph embedding. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 18, pages 4396–4402, 2018.

[36] Zequn Sun, Jiacheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference (ISWC)*, pages 612–629, 2019.

[37] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 222–229, 2020.

[38] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment*, 13(12), 2020.

[39] Anil Surisetty, Deepak Chaurasiya, Nitish Kumar, Alok Singh, Gaurav Dhama, Aakarsh Malhotra, Ankur Arora, and Vikrant Dey. Reps: Relation, position and structure aware entity alignment. In *Companion Proceedings of the Web Conference 2022 (WWW)*, pages 1083–1091, 2022.

[40] Zhiquan Tan, Yifan Zhang, Jingqin Yang, and Yang Yuan. Contrastive learning is spectral clustering on similarity graph, 2023.

[41] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 349–357, 2018.

[42] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. Cross-lingual knowledge graph alignment via graph matching neural network. In *Conference of the Association for Computational Linguistics (ACL)*, pages 3156–3161, 2019.

[43] Jaemin Yoo, Junghun Kim, Hoyoung Yoon, Geonsoo Kim, Changwon Jang, and U Kang. Graph-based pu learning for binary and multiclass classification without class prior. *Knowledge and Information Systems*, 64(8):2141–2169, 2022.

 https://doi.org/10.52202/079017-2507

[44] Ziheng Zhang, Hualuo Liu, Jiaoyan Chen, Xi Chen, Bo Liu, Yuejia Xiang, and Yefeng Zheng. An industry evaluation of embedding-based entity alignment. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 179–189, 2020.

Appendix

A Notation

A.1 Definitions

Definition 1 (Knowledge graph)**.** *Knowledge graph (KG) is a directed graph $G = (E, R, T)$ comprising three distinct sets: entities E, relations R, and triples $T \subseteq E \times R \times E$. KG is stored in the form of triples <entity, relation, entity>, with entities denoted by nodes and the relation between entities defined by edges.*

Definition 2 (Entity alignment)**.** *Given source KG and target KG, corresponding to $G_s = (E_s, R_s, T_s)$ and $G_t = (E_t, R_t, T_t)$ respectively, and $A = \{(u, v)|u \in E_s, v \in E_t, u \equiv v\}$ a set of pre-aligned anchor node pairs, where $\equiv$ indicates equivalence, the goal of entity alignment is to identify additional pairs of potentially equivalent entities using information from G_s, G_t, and A. This task typically assumes a one-to-one correspondence between E_s and E_t.*

Definition 3 (Entity alignment with dangling cases)**.** *Let entities in the source and target graphs be composed of two types of nodes: $E_s = D_s \cup M_s, E_t = D_t \cup M_t$, where D_s, D_t denote dangling sets that contain entities that have no counterparts, and M_s, M_t are matchable sets. A set of pre-aligned anchor node pairs are $S = \{(u, v)|u \in M_s, v \in M_t, u \equiv v\}$. The task seeks to discover the remaining aligned entities given G_s, G_t, and S.*

A.2 Transductive Learning:

Transductive learning models are trained from observed, specific (training) cases to specific (test) cases, employing both training and test information except for test labels. In contrast, the inductive learning model is reasoning from observed training cases to general rules, which are then applied to the test cases. Let's get down to the EA task. As KG structure is accessible through given triples, which can accurately describe the connections between entities. When we attempt to figure out entity alignment tasks, the KG structure information of the whole source and target KGs is what we could exploit, covering potentially (test) equivalent entities' relative positions. That's why we confine the problem field to transductive learning.

A.3 Graph Convolutional Networks

For graph convolutional networks (GCN) [12], the embedding $\boldsymbol{h}_{e_i}^{l+1}$ of node e_i at the $l+1$-th layer is updated iteratively by aggregating node features of the neighboring nodes $\mathcal{N}_{e_i}$ from the prior layer:

$$\boldsymbol{h}_{e_i}^{l+1} = \sigma\left(\sum_{e_j \in \mathcal{N}_{e_i} \cup \{e_i\}} \alpha_{i,j} W^{l+1} \boldsymbol{h}_{e_j}^{l}\right), \tag{7}$$

where each embedding $\boldsymbol{h}_{e_i}^{l}$ represents the d-dimensional embedding vector of e_i, $\alpha_{i,j}$ denotes the weight coefficient between e_i and e_j, W^{l+1} being the transformation matrix of the $(l+1)$-th GNN layer, and σ being the activation function.

B Proof for Lemma 2

Proof. The $\mathcal{L}_{\text{info}}$ also has a good effect on mining high-quality negative samples, which we show has an equivalent effect to *truncated uniform negative sampling (TUNS)* in [35]. TUNS points out that negative samples obtained by uniform random sampling are highly redundant since only high-quality negative samples improve the model. Thus TUNS chooses the K-nearest neighbors of the e_i as the negative samples, which are most challenging to distinguish. In the special case of $K = 1$, the loss can be written as $\mathcal{L}_{\text{TUNS}} = \sum_{e_i \in \mathcal{X}_p} \max_j(H(e_i, e_+^i, e_j^i))$. If we approximate the $\max$ function by the LogSumExp, the contrastive loss function turns to

$$\mathcal{L}_{\text{TUNS}} \approx \sum_{e_i \in \mathcal{X}_p} \frac{1}{\lambda} \log\left(\sum_{j}^{N} \exp(\lambda\, H(e_i, e_+^i, e_j^i))\right), \tag{8}$$

https://doi.org/10.52202/079017-2507

minimizing which is equivalent to minimizing Eq. (3). Hence our contrastive learning loss is actually a special form of TUNS. For randomly sampled negative samples, $\mathcal{L}_{\text{info}}$ can play a role in preferentially optimizing high-quality negative samples.

□

C Proof for Theorem 1

Proof. The risk of g is $R(g) = \mathbb{E}_{(X,Y)\sim p(x,y)}[\ell(g(X), Y)] = \pi_p R_{\mathrm{p}}^{+}(g) + \pi_n R_{\mathrm{n}}^{-}(g)$ in positive negative learning problems. In positive-unlabeled learning where $\mathcal{X}_{\mathrm{n}}$ is unavailable, we can only approximate $R(g)$ by positive samples and unlabeled samples. We represent the unlabeled distribution as $p_{\mathrm{u}}(x) = \pi_{\mathrm{n}}^{\mathrm{u}} p_{\mathrm{n}}(x) + \pi_{\mathrm{p}}^{\mathrm{u}} p_{\mathrm{p}}(x)$, so that the negative distribution can be written as $\pi_{\mathrm{n}} p_{\mathrm{n}}(x) = \frac{\pi_{\mathrm{n}}}{\pi_{\mathrm{n}}^{u}} \cdot \left[p_{\mathrm{u}}(x) - \pi_{\mathrm{p}}^{u} p_{\mathrm{p}}(x)\right]$. Provided $R_{\mathrm{p}}^{-}(g) = \mathbb{E}_{X\sim p_{\mathrm{p}}(x)}[\ell(g(X), -1)]$ and $R_{\mathrm{u}}^{-}(g) = \mathbb{E}_{X\sim p_{\mathrm{u}}(x)}[\ell(g(X), -1)]$, we obtain that

$$\pi_{\mathrm{n}} R_{\mathrm{n}}^{-}(g) = \frac{\pi_{\mathrm{n}}}{\pi_{\mathrm{n}}^{\mathrm{u}}} \cdot \left[R_{\mathrm{u}}^{-}(g) - \pi_{\mathrm{p}}^{\mathrm{u}} R_{\mathrm{p}}^{-}(g)\right], \tag{9}$$

and

$$\widehat{R}_{\mathrm{pu}}(g) = \pi_{\mathrm{p}} \widehat{R}_{\mathrm{p}}^{+}(g) + \frac{\pi_{\mathrm{n}}}{\pi_{\mathrm{n}}^{\mathrm{u}}} \cdot \left[\widehat{R}_{\mathrm{u}}^{-}(g) - \pi_{\mathrm{p}}^{\mathrm{u}} \widehat{R}_{\mathrm{p}}^{-}(g)\right]. \tag{10}$$

Specifically, $\pi_{\mathrm{n}} = 1 - \pi_{\mathrm{p}}$, $\pi_{\mathrm{n}}^{\mathrm{u}} = 1 - \pi_{\mathrm{p}}^{\mathrm{u}}$ could be derived as *class-prior probability given* π_{p} *and* $\pi_{\mathrm{p}}^{\mathrm{u}}$. In particular, the ratio of labeled positive samples could be precisely figured out as π_{p}^{tr} in transductive learning, given which $\pi_{\mathrm{p}}^{\mathrm{u}} = \frac{\pi_{\mathrm{p}} - \pi_{\mathrm{p}}^{tr}}{1 - \pi_{\mathrm{p}}^{tr}}$, $\pi_{\mathrm{n}}^{\mathrm{u}} = 1 - \pi_{\mathrm{p}}^{\mathrm{u}}$ could be derived as *class-prior probability*.

□

D Proof for Theorem 2

Proof. $\mathfrak{R}_{n,q}$ is defined as the *Rademacher complexity* of the class of classifiers $\mathcal{G}$ for the sampling of size n from distribution $q(x)$. From [30] we have that with probability at least $1 - \delta/2$, the uniform deviation bounds below hold separately:

$$\begin{aligned}
\sup_{g\in\mathcal{G}} |\widehat{R}_{+}(g) - R_{+}(g)| &\leq 2L_{\ell}\mathfrak{R}_{n_{+},p_{\mathrm{p}}}(\mathcal{G}) + \sqrt{\frac{\ln(4/\delta)}{2n_{+}}} \\
&\triangleq M_{+} > 0, \\
\sup_{g\in\mathcal{G}} |\widehat{R}_{\mathrm{u},-}(g) - R_{\mathrm{u},-}(g)| &\leq 2L_{\ell}\mathfrak{R}_{n_{\mathrm{u}},p_{\mathrm{u}}}(\mathcal{G}) + \sqrt{\frac{\ln(4/\delta)}{2n_{\mathrm{u}}}} \\
&\triangleq M_{-} > 0,
\end{aligned}$$

where L_{ℓ} is the *Lipschitz constant* of loss ℓ in its first parameter and n_{+} is the number of positive samples while n_u is that of unlabeled. Following the *symmetric condition assumption* in [30], it is obviously holds that:

$$\begin{aligned}
\mathrm{Var}(\widehat{R'}_{\mathrm{pu}}(g)) &= 2\pi_{\mathrm{p}} M_{+} + M_{-}, \text{while} \\
\mathrm{Var}(\widehat{R}_{\mathrm{pu}}(g)) &= (\pi_{\mathrm{p}} + \frac{\pi_{\mathrm{n}} \cdot \pi_{\mathrm{p}}^{\mathrm{u}}}{\pi_{\mathrm{n}}^{\mathrm{u}}}) M_{+} + \frac{\pi_{\mathrm{n}}}{\pi_{\mathrm{n}}^{\mathrm{u}}} M_{-}
\end{aligned}$$

holds in our setting. Then it is evident that $\frac{\pi_{\mathrm{n}}}{\pi_{\mathrm{p}}} < \frac{\pi_{\mathrm{n}}^{u}}{\pi_{\mathrm{p}}^{u}}$, i.e., $\frac{\pi_{\mathrm{n}} \cdot \pi_{\mathrm{p}}^{\mathrm{u}}}{\pi_{\mathrm{n}}^{\mathrm{u}}} < \pi_{\mathrm{p}}$ and $\frac{\pi_{\mathrm{n}}}{\pi_{\mathrm{n}}^{u}} < 1$.

Consequently, comparing the coefficients of M_{+} and M_{-} leads to the conclusion that $\widehat{R}_{\mathrm{pu}}(g)$ could possess tighter uniform deviation bound than that of *Non-negative Risk Estimator* [13].

□

E Convergence Proof

The expectation–maximization (EM) algorithm is an iterative approach to maximize the likelihood $p(\mathbf{y}|\mathbf{X};\theta)$ of target variables y over input variables $\mathbf{X}$ and parameters θ. The EM algorithm works iteratively, and each iteration consists of an expectation (E) step and a maximization (M) step. The E step computes the expectation of the log-likelihood concerning the conditional distribution of the latent variable $\mathbf{z}$ given the current parameters $\theta^{(t)}$ at the t-th iteration:

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}|\mathbf{X},\mathbf{y},\theta^{(t)})}[\log p(\mathbf{y},\mathbf{z} \mid \mathbf{X},\theta)]. \tag{11}$$

Then, the M step finds a set of parameters that maximizes the computed expectation:

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta \mid \theta^{(t)}). \tag{12}$$

Due to the maximization step, we get the following inequality naturally:

$$Q(\theta^{(t+1)} \mid \theta^{(t)}) - Q(\theta^{(t)} \mid \theta^{(t)}) \geq 0. \tag{13}$$

Lemma 3 (Convergence of EM algorithm [29])**.** *It is guaranteed that the EM algorithm always improves* $\log p(\mathbf{y} \mid \mathbf{X},\theta)$ *by increasing the value of* $Q(\theta \mid \theta^{(t)})$*. Since* $\log p(\mathbf{y} \mid \mathbf{X},\theta)$ *is monotonically bounded, the EM must converge.*

Proof. The following equation holds for any z:

$$\log p(\mathbf{y} \mid \mathbf{X},\theta) = \log p(\mathbf{y},\mathbf{z} \mid \mathbf{X},\theta) - \log p(\mathbf{z} \mid \mathbf{X},\mathbf{y},\theta).$$

We take the expectation over $p(\mathbf{z} \mid \mathbf{X},\mathbf{y},\theta^{(t)})$ for both sides as follows:

$$\begin{aligned}
&\mathbb{E}_{p(\mathbf{z}|\mathbf{X},\mathbf{y},\theta^{(t)})}[\log p(\mathbf{y} \mid \mathbf{X},\theta)] \\
&= \mathbb{E}_{p(\mathbf{z}|\mathbf{X},\mathbf{y},\theta^{(t)})}[\log p(\mathbf{y},\mathbf{z} \mid \mathbf{X},\theta)] - \mathbb{E}_{p(\mathbf{z}|\mathbf{X},\mathbf{y},\theta^{(t)})}[\log p(\mathbf{z} \mid \mathbf{X},\mathbf{y},\theta)] \\
&= Q(\theta \mid \theta^{(t)}) + H(p(\mathbf{z} \mid \mathbf{X},\mathbf{y},\theta) \mid p(\mathbf{z} \mid \mathbf{X},\mathbf{y},\theta^{(t)})) \\
&\triangleq Q(\theta \mid \theta^{(t)}) + H(p_{\theta} \mid p_{\theta^{(t)}})
\end{aligned}$$

where H stands for the entropy. If we substitute $\theta^{(t)}$ for θ, we get the following:

$$\log p(\mathbf{y} \mid \mathbf{X},\theta^{(t)}) = Q(\theta^{(t)} \mid \theta^{(t)}) + H(p_{\theta^{(t)}} \mid p_{\theta^{(t)}}). \tag{14}$$

Gibbs' inequality states that $H(q \mid p) - H(p \mid p) \geq 0$ always holds for any distribution p and q. Hence we have:

$$\begin{aligned}
&\log p(\mathbf{y} \mid \mathbf{X},\theta^{(t+1)}) - \log p(\mathbf{y} \mid \mathbf{X},\theta^{(t)}) \\
&= Q(\theta^{(t+1)} \mid \theta^{(t)}) + H(p_{\theta^{(t+1)}} \mid p_{\theta^{(t)}}) - Q(\theta^{(t)} \mid \theta^{(t)}) - H(p_{\theta^{(t)}} \mid p_{\theta^{(t)}}) \\
&= Q(\theta^{(t+1)} \mid \theta^{(t)}) - Q(\theta^{(t)} \mid \theta^{(t)}) + H(p_{\theta^{(t+1)}} \mid p_{\theta^{(t)}}) - H(p_{\theta^{(t)}} \mid p_{\theta^{(t)}}) \\
&\geq Q(\theta^{(t+1)} \mid \theta^{(t)}) - Q(\theta^{(t)} \mid \theta^{(t)}) = Q(\theta^{(t+1)} \mid \theta^{(t)}) - Q(\theta^{(t)}) \geq 0.
\end{aligned}$$

□

Now we provide the proof for Theorem 3.

Proof. According to Lemma 3, we have that the EM algorithm converges. Next, we model our problem and give the approximate equivalence between our algorithm and the EM algorithm to prove our algorithm converges.

The latent variables $\mathbf{z}$ represent the true label distribution of unlabeled samples, where $\hat{y}_i(u) = f(\mathbf{X},i;\theta)$ is the probability of node i being labeled as $u \in \{+1,-1\}$ by the current classifier f. Given $\pi_{\mathrm{p}}^{\mathrm{u}}$, the label distribution of all unlabeled samples is:

$$p(z_i) = \begin{cases} \hat{\pi}_{\mathrm{p}}^{\mathrm{u}} & \text{if} \quad z_i = +1, \\ 1 - \hat{\pi}_{\mathrm{p}}^{\mathrm{u}} & \text{if} \quad z_i = -1. \end{cases} \tag{15}$$

https://doi.org/10.52202/079017-2507

First, the conditional distribution $p(\mathbf{z} \mid \mathbf{X}, \mathbf{y}, \theta^{(t)})$ of latent variables given the current parameters $\theta^{(t)}$ is approximated by:

$$p(\mathbf{z} \mid \mathbf{X}, \mathbf{y}, \theta^{(t)}) \approx \prod_{i \in \mathcal{U}} p(z_i). \tag{16}$$

Second, the joint distribution $p(\mathbf{y}, \mathbf{z} \mid \mathbf{X}, \theta)$ of labeled and unlabeled nodes with new parameters θ is approximated by the classifier f, which is also considered as a marginalization function that gives the label distribution of each node based on all given information:

$$p(\mathbf{y}, \mathbf{z} \mid \mathbf{X}, \theta) \approx \prod_{i \in \mathcal{P}} \hat{y}_i(+1) \prod_{j \in \mathcal{U}} \hat{y}_j(z_j) \tag{17}$$

We propose to use the average log-likelihood differences to measure the classification preference of the model. It can be transformed into the following form by $\Delta_{\mathcal{U}}$ and $\Delta_{\mathcal{P}}$, representing that on domain $\mathcal{U}$ and $\mathcal{P}$:

$$\Delta_{\mathcal{U}} = \frac{1}{|\mathcal{U}|} \sum_{j \in \mathcal{U}} \log \hat{y}_j(+1) - \log \hat{y}_j(-1)$$

$$\Delta_{\mathcal{P}} = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \log \hat{y}_i(+1) - \log \hat{y}_i(-1)$$

If the mean value is positive, it means that the logarithmic probability of positive classes is higher than that of negative classes in the given domain, and vice versa. When we ignore the preference of the classification model, this actually describes the category feature bias in certain domains.

Let's rethink the meaning of ***preference condition***. If the model has a similar classification preference in domain $\mathcal{U}$ and $\mathcal{P}$, we can express as $\Delta_{\mathcal{U}} \approx \Delta_{\mathcal{P}}$. This condition can be arranged as the mathematical expression of the condition given by Theorem 3 through the properties of the $\log$ function.

$$\Delta_{\mathcal{U}} \approx \Delta_{\mathcal{P}} \equiv \sum_{j \in \mathcal{U}} \frac{1}{|\mathcal{U}|} \log \frac{\hat{y}_j(+1)}{\hat{y}_j(-1)} \approx \sum_{i \in \mathcal{P}} \frac{1}{|\mathcal{P}|} \log \frac{\hat{y}_i(+1)}{\hat{y}_i(-1)}$$

We derive $-\mathcal{L}_{\mathrm{pu}}$ from Eq.(7), since the goal of training is to minimize the objective function:

$$\begin{aligned}
&\frac{1}{N} \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{X}, \mathbf{y}, \theta^{(t)})} [\log p(\mathbf{y}, \mathbf{z} \mid \mathbf{X}, \theta)] \\
&= \frac{1}{N} \sum_{\mathbf{z}} p(\mathbf{z} \mid \mathbf{X}, \mathbf{y}, \theta^{(t)}) \log p(\mathbf{y}, \mathbf{z} \mid \mathbf{X}, \theta) \\
&\approx \frac{1}{N} \sum_{\mathbf{z}} p(\mathbf{z} \mid \mathbf{X}, \mathbf{y}, \theta^{(t)}) (\sum_{i \in \mathcal{P}} \log \hat{y}_i(+1) + \sum_{j \in \mathcal{U}} \log \hat{y}_j(z_j)) \\
&= \frac{1}{N} \sum_{i \in \mathcal{P}} \log \hat{y}_i(+1) + \frac{1}{N} \sum_{j \in \mathcal{U}} \sum_{z_j \in \pm 1} p(z_j) \log \hat{y}_j(z_j) \\
&= \frac{|\mathcal{P}|}{N} \cdot \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \log \hat{y}_i(+1) + \frac{|\mathcal{U}|}{N} \cdot \frac{1}{|\mathcal{U}|} \sum_{j \in \mathcal{U}} \left(\hat{\pi}_{\mathrm{p}}^{\mathrm{u}} \log \hat{y}_j(+1) + (1 - \hat{\pi}_{\mathrm{p}}^{\mathrm{u}}) \log \hat{y}_j(-1) \right) \\
&= \frac{|\mathcal{P}|}{N} \cdot \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \log \hat{y}_i(+1) + \frac{|\mathcal{U}|}{N} \cdot \frac{1}{|\mathcal{U}|} \sum_{j \in \mathcal{U}} \log \hat{y}_j(-1) + \frac{\hat{\pi}_{\mathrm{p}}^{\mathrm{u}} |\mathcal{U}|}{N} \Delta_{\mathcal{U}}
\end{aligned}$$

We replaced $\Delta_{\mathcal{U}}$ with $\Delta_{\mathcal{P}}$, and this process is completed by the above ***preference condition***.

$$\begin{aligned}
&\approx \frac{|\mathcal{P}|}{N} \cdot \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \log \hat{y}_i(+1) + \frac{|\mathcal{U}|}{N} \cdot \frac{1}{|\mathcal{U}|} \sum_{j \in \mathcal{U}} \log \hat{y}_j(-1) + \frac{|\mathcal{U}| \hat{\pi}_{\mathrm{p}}^{\mathrm{u}}}{N} \cdot \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \log \hat{y}_i(+1) - \log \hat{y}_i(-1) \\
&= -\frac{|\mathcal{P}| + |\mathcal{U}| \cdot \hat{\pi}_{\mathrm{p}}^{\mathrm{u}}}{|\mathcal{P}| + |\mathcal{U}|} \sum_{i \in \mathcal{P}} \frac{1}{|\mathcal{P}|} \log \hat{y}_i(+1) - \frac{\frac{|\mathcal{N}|}{|\mathcal{P}| + |\mathcal{U}|}}{\frac{|\mathcal{N}|}{|\mathcal{U}|}} \left(\sum_{j \in \mathcal{U}} \frac{1}{|\mathcal{U}|} \log \hat{y}_j(-1) \ - \hat{\pi}_{\mathrm{p}}^{\mathrm{u}} \sum_{i \in \mathcal{P}} \frac{1}{|\mathcal{P}|} \log \hat{y}_i(-1) \right)
\end{aligned}$$

Datasets		# Entities	# Rel.	# Triples	# Dang	# Align
$\mathrm{DBP2.0}_{\mathrm{ZH-EN}}$	Chinese	84,996	3,706	286,067	51,813	33,183
	English	118,996	3,402	586,868	85,813	
$\mathrm{DBP2.0}_{\mathrm{JA-EN}}$	Japanese	100,860	3,243	347,204	61,090	39,770
	English	139,304	3,396	668,341	99,534	
$\mathrm{DBP2.0}_{\mathrm{FR-EN}}$	French	221,327	2,841	802,678	97,375	123,952
	English	278,411	4,598	1,287,231	154,459	
$\mathrm{DBP15K}_{\mathrm{ZH-EN}}$	Chinese	19,388	1,701	70,414	4,388	15,000
	English	19,572	1,323	95,142	4,572	
$\mathrm{DBP15K}_{\mathrm{JA-EN}}$	Japanese	19,814	1,299	77,214	4,814	15,000
	English	19,780	1,153	93,484	4,780	
$\mathrm{DBP15K}_{\mathrm{FR-EN}}$	French	19,661	903	105,998	4,661	15,000
	English	19,993	1,208	115,722	4,993	
GA16K	None	6,208	8	68,534	0	6,208
	None	16,363	12	151,662	10,155	

Table 5: Statistics of DBP2.0, DBP15K and GA16K.

Datasets		# Entities	# Rel.	# Triples	# Dang	# Align
$\mathrm{DBP2.0}_{\mathrm{ZH-EN}}\mathrm{Plus}$	Chinese	69,386	3,455	241,588	36,302	33,084
	English	94,026	3,131	470,284	60,942	
$\mathrm{DBP2.0}_{\mathrm{JA-EN}}\mathrm{Plus}$	Japanese	82,192	3,011	291,406	42,588	39,604
	English	110,362	3,054	532,988	70,758	
$\mathrm{DBP2.0}_{\mathrm{ZH-EN}}\mathrm{Minus}$	Chinese	72,252	3,351	200,400	54,594	17,658
	English	107,853	3,140	421,597	90,195	
$\mathrm{DBP2.0}_{\mathrm{JA-EN}}\mathrm{Minus}$	Japanese	86,241	3,014	236,546	64,841	21,400
	English	126,558	3,166	485,133	105,158	

Table 6: Statistics of DBP2.0-Plus and DBP2.0-Minus

Denote $\pi_{\mathrm{p}} = \frac{|\mathcal{P}|+|\mathcal{U}|\cdot\hat{\pi}_{\mathrm{p}}^{\mathrm{u}}}{|\mathcal{P}|+|\mathcal{U}|}$, $\pi_{\mathrm{n}} = \frac{|\mathcal{N}|}{|\mathcal{P}|+|\mathcal{U}|}$ and $\pi_{\mathrm{n}}^{\mathrm{u}} = \frac{|\mathcal{N}|}{|\mathcal{U}|}$. The above formula is equivalent to:

$$\begin{aligned}
&\arg\max_{\theta} \frac{1}{N}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}|\mathbf{X},\mathbf{y},\theta^{(t)})}[\log p(\mathbf{y},\mathbf{z}\mid\mathbf{X},\theta)] \\
&= \arg\max_{\theta} \underbrace{-\pi_{\mathrm{p}}\widehat{R}_{\mathrm{p}}^{+}(g) - \frac{\pi_{\mathrm{n}}}{\pi_{\mathrm{n}}^{\mathrm{u}}}\cdot\left[\widehat{R}_{\mathrm{u}}^{-}(g) - \pi_{\mathrm{p}}^{\mathrm{u}}\widehat{R}_{\mathrm{p}}^{-}(g)\right]}_{-\widehat{R}_{\mathrm{pu}}(g)} \\
&= \arg\min_{\theta} \widehat{R}_{\mathrm{pu}}(g) = \arg\max_{\theta} -\mathcal{L}_{\mathrm{pu}}
\end{aligned} \tag{18}$$

□

F Statistics of Experimental Dataset and Baselines

Datasets. The training/test sets for each dataset are generated using a fixed random seed. For entity alignment, 30% of matchable entity pairs constitute the training set, while the remaining form the test set. For dangling entity detection, we did not utilize any labeled dangling entity data, in contrast to prior work which labels 30% of the dangling entities and matchable pairs respectively for training [33]. Hence our method imposes minimal restrictions on annotated data. All datasets are briefly introduced in the following and some statistics are provided in Tab. 7.

In addition to the existing datasets, we also constructed DBP2.0-minus & -plus as supplementary to DBP2.0, GA16K enabling comparison between Dangling-Entities-Unaware baselines, and GA-DBP15K for evaluation of iPULE.

DBP15K[3] [34]: DBP15K consists of three cross-lingual subsets constructed from DBpedia: English-French($\mathrm{DBP}_{\mathrm{FR-EN}}$), English-Chinese ($\mathrm{DBP}_{\mathrm{ZH-EN}}$), English-Japanese($\mathrm{DBP}_{\mathrm{JA-EN}}$). Each subset

[3]https://paperswithcode.com/dataset/dbp15k

https://doi.org/10.52202/079017-2507

GA-DBP15K		Entities	Dang	Align
$\mathrm{GA-EN}$	GA EN-share	16,363 19,388 + Align	16,363 - Align 19,388	16,363$*$c%
$\mathrm{GA-ZH}$	GA ZH-share	16,363 19,572 + Align	16,363 - Align 19,572	16,363$*$c%
$\mathrm{GA-JA}$	GA JA-share	16,363 19,814 + Align	16,363 - Align 19,814	16,363$*$c%
$\mathrm{GA-FR}$	GA FR-share	19,388 19,661 + Align	16,363 - Align 19,661	16,363$*$c%

Table 7: Statistics of GA-DBP15K. c = [25%,20%,15%,10%].

contains 15,000 pre-aligned entity pairs. This dataset includes a small proportion of dangling entity samples which is yet mostly ignored in previous entity alignment tasks.

DBP2.0[4] [33]: DBP2.0 is an entity alignment dataset with a considerable proportion of dangling entities, constructed from the multilingual Infobox Data of DBpedia [2]. The dataset contains three pairs of crosslingual KGs, ZH-EN (Chinese to English), JA-EN (Japanese to English), and FR-EN (French to English). Since there are dangling nodes in both the source and target graphs, we separately test source-to-target and target-to-source alignment, consistent with the established approach. A representative feature of the dataset is that the matchable and dangling entities exhibit similar degree distributions which are hard to distinguish, displaying a real-world challenge in aligning knowledge graphs. Based on DBP2.0, we extend the following -minus & -plus datasets for verification of iPULE on different positive proportions between 20%-50%.

DBP2.0-plus: In the construction of the plus dataset, our goal is to construct the dataset that has a higher π_p, and we realize this by reducing a few existing dangling entities on ZH-EN and JA-EN. We randomly delete dangling entities from both source and target KG equally and remove triples containing them. The constructed DBP2.0-plus are reindexed and thus obtain a higher π_{p} value than the original dataset.

DBP2.0-minus: In contrast, to lower the π_{p} value. Given the constraint of preventing new dangling entities that could introduce false information to the KG, we can only reduce the number of matchable entities. Given source and target KG, removing one entity from a pair makes the remaining entity dangling. We randomly delete matchable entities from one side of the pair on both source and target KG uniformly. The constructed DBP2.0-minus are reindexed and thus obtain a lower π_{p} value than the original dataset.

GA16K: This dataset constructed by us exclusively contains dangling nodes in the target graph, facilitating a comparison between our work and baselines that neglect dangling entities. GA16K is extracted from GAKG[5] [7], a Geoscience Academic Knowledge Graph. We first order each type of entity in GAKG according to their degrees and select the entities with a large degree into the entity set. A total of 16,363(16K) separate entities and their relations were extracted to compose the target graph. Then we extract 6,208 entities from the target graph to comprise the source graph. Hence there are 6,208 ground-truth matchable pairs between the source and the target. The remaining 10,155 entities in the target graph are regarded as dangling entities.

GA-DBP15K: The GA-DBP15K dataset is derived from a subset of entities within GA16K, along with their associated triples, which are then concatenated with the DBP15K dataset, such as EN, resulting in a new dataset pair that shares a proportion of common entities. To achieve the goal, we first extract a certain proportion of triples from GA16K. We then reindex all the entities from the extricated GA16K and DBP15K datasets. Finally, we update the entity and relation indices in the triples, replacing them with the newly assigned indices.

Baselines. Since our work does not take advantage of any side information, we emphasize its comparison with the previous methods purely depending on graph structures. These works majorly incorporate two types:

Dangling-Entities-Unaware. We include advanced entity alignment methods in recent years: GCN-Align [41], RSNs [9], MuGNN [4], KECG [16]. Methods with bootstrapping to generate semi-

[4]https://github.com/nju-websoft/OpenEA/tree/master/dbp2.0

[5]https://github.com/davendw49/gakg

supervised structure data are also adopted: BootEA [35], TransEdge [36], MRAEA [26], AliNet [37], and Dual-AMN [25].

Dangling-Entities-Aware. To the best of our knowledge, the method of [33] is the most fairly comparable baseline which is based on MTransE [5] and AliNet [37]. Because MHP [19] over-emphasized more use of labeled dangling data like high-order similarity information which is also based on the above two methods, while SoTead [22] and UED [24] utilize additional side-information. SoTead [22] and UED [24] can only execute the degraded version on DBP2.0 cause no side-information is available on that. We exclude them from baselines for our methods. [33] introduces three techniques to address the dangling entity issue: nearest neighbor (NN) classification, marginal ranking (MR), and background ranking (BR).

Metrics are set for the dangling entity detection task and the entity alignment task separately. For the entity detection, we evaluate the detection performance by the standard precision, recall, and F1 score. To align the previous dangling detection baselines, we detect dangling entities as 'positive' samples and align matchable entities for entity alignment.

For the entity alignment, the metrics slightly differ in the dangling-entities-unaware and dangling-entities-aware settings. We evaluate the baselines unaware of the dangling entities by following their assumptions and using their metric Hits@K (K= $1, 10, 50$, H@K for short) on the ranking list S. This setting is referred to as *relaxed setting* when S is composed of all ground-truth entities without the dangling ones:

$$Hits@K = \frac{1}{|S|}\sum_{k=1}^{|S|} \mathbb{1}(\text{rank}_i \leq k).$$

In contrast, we refer to a *consolidated setting* for baselines aware of dangling entities. In this setting, the ranking list S also contains all dangling entities. We use H@K in the consolidated setting to evaluate the performance of baselines aware of but not removing dangling entities in the alignment. For baselines where dangling entities are detected and removed before alignment, the direct use of H@K to evaluate entity alignment may not be precise, since errors are introduced in the detection phase. Thus we follow the convention of [33] to apply a set of metrics for evaluating the accuracy of entity alignment in the consolidated setting. Each of them is derived and introduced as follows.

The standard precision and recall is given as

$$\text{precision} = \frac{TP}{TP+FP}, \;\; \text{recall} = \frac{TP}{TP+FN}$$

for dangling entity detection. We denote the dangling entities as 0 and matchable ones as 1. The subscript t_{1y} suggests that an entity with ground truth y is classified as matchable. Likewise, t_{y1} represents a matchable entity that is classified as y by the detection classifier. If a source entity is dangling but not identified by the detection module, its alignment result is always considered incorrect, i.e., $H@K_{t_{10}} = 0$. Hence we have the precision for entity alignment as

$$\begin{aligned} H@1_{t_{1y}} &= \text{precision} \cdot H@1_{t_{11}} + (1 - \text{precision}) \cdot H@1_{t_{10}} \\ &= \text{precision} \cdot H@1_{t_{11}}. \end{aligned} \tag{19}$$

Similarly, if a matchable entity is falsely excluded by the dangling detection module, this test case is also regarded as incorrect $H@K_{t_{01}} = 0$ since the alignment model has no chance to search for alignment. Hence we have the recall for entity alignment as

$$\begin{aligned} H@1_{t_{y1}} &= \text{recall} \cdot H@1_{t_{11}} + (1 - \text{recall}) \cdot H@1_{t_{01}} \\ &= \text{recall} \cdot H@1_{t_{11}}. \end{aligned} \tag{20}$$

For the methods that are aware of dangling entities, we use $H@1_{t_{1y}}$ and $H@1_{t_{y1}}$ to denote the precision and recall of the entity alignment task. Similarly, we define the F1 score of the entity alignment as the harmonic average of precision and recall:

$$F1 = \frac{2 \cdot H@1_{t_{1y}} \cdot H@1_{t_{y1}}}{H@1_{t_{1y}} + H@1_{t_{y1}}}. \tag{21}$$

Later, $H@1_{t_{1y}}$ and $H@1_{t_{y1}}$ are referred to as Prec. and Rec. in reporting alignment performance.

 https://doi.org/10.52202/079017-2507

G Additional Experiment

RQ1: How do current network alignment methods perform in unlabeled dangling cases? (see appendix G.1)

RQ2: Loss convergence on GA-DBP15K and DBP2.0. (see appendix G.2)

RQ3: How do we select the embedding dimensions? (see appendix G.3)

RQ4: What is the actual efficiency of our approach? (see appendix G.4)

RQ5: Baseline comparison under different pre-aligned seeds? (see appendix G.5)

RQ6: Additional experiments involved LightEA as a strong baseline? (see appendix G.6)

G.1 The Non-Negligibility of dangling Problem (RQ1).

We investigated the performance degradation of various existing EA methods in the face of the dangling problem, which shows that this problem is worth considering.

Method	$\text{DBP15K}_{\text{ZH-EN}}$			$\text{DBP15K}_{\text{JA-EN}}$			$\text{DBP15K}_{\text{FR-EN}}$		
	H@1	H@10	H@50	H@1	H@10	H@50	H@1	H@10	H@50
BootEA	31.30↓ 20.96	59.70↓ 16.18	71.51↓ 12.91	33.77↓ 15.27	62.66↓ 11.64	73.09↓ 10.29	23.11↓ 26.72	58.39↓ 18.77	71.54↓ 14.00
TransEdge	49.91↓ 15.21	76.62↓ 9.79	83.44↓ 7.16	54.07↓ 13.42	78.01↓ 8.25	84.00↓ 6.21	48.23↓ 17.34	79.32↓ 9.70	86.69↓ 6.24
MRAEA	59.45↓ 5.62	83.04↓ 2.53	88.68↓ 1.56	61.60↓ 4.45	83.48↓ 2.21	88.65↓ 1.50	61.55↓ 6.62	85.85↓ 2.61	90.79↓ 1.69
GCN-Align	31.99↓ 10.70	62.21↓ 6.45	71.93↓ 4.31	32.08↓ 10.08	61.04↓ 5.86	70.34↓ 3.52	30.71↓ 10.50	61.64↓ 7.07	72.45↓ 5.55
RSNs	43.00↓ 8.50	62.90↓ 8.00	69.70↓ 7.00	20.60↓ 31.60	44.60↓ 26.60	53.20↓ 23.60	36.30↓ 15.30	63.30↓ 10.10	71.70↓ 7.80
MuGNN	34.66↓ 14.75	68.48↓ 9.32	80.53↓ 5.69	32.93↓ 14.68	66.68↓ 8.82	78.63↓ 5.67	34.93↓ 14.02	68.88↓ 9.69	81.67↓ 5.32
KECG	35.92↓ 12.87	65.70↓ 10.35	76.44↓ 8.06	32.31↓ 15.48	63.19↓ 11.96	74.42↓ 9.29	32.84↓ 15.47	64.78↓ 11.98	76.70↓ 8.35
AliNet	53.84↓ 0.66	73.73↓ 3.16	80.30↓ 1.59	52.69↓ 1.30	74.01↓ 2.60	80.91↓ 1.90	54.01↓ 0.58	76.19↓ 2.74	83.25↓ 1.40
Dual-AMN	60.72↓ 12.20	83.93↓ 5.22	89.45↓ 3.54	62.29↓ 10.62	83.38↓ 5.35	88.80↓ 3.21	65.33↓ 10.48	87.76↓ 4.17	92.47↓ 2.24

Table 8: Network alignment performance on DBP15K in the consolidated setting. The blue numbers suggest the drop from the relaxed setting (as with their original implementation).

We reproduce the baselines unaware of dangling entities on DBP15K in the relaxed setting. On the same dataset, we rerun their methods but in a consolidated setting that takes the dangling entities into account. Even though DBP15K only comprises a small percentage of dangling entities, the drop in the consolidated setting is significant, as shown in Tab. 8.

The reason behind such a performance drop is mainly because most previous works remove dangling entities from the ground truth in measuring their alignment performance. In particular, Dual-AMN takes advantage of the bootstrapping module by incorporating labeled pairs in training. In the relaxed setting, such labeled pairs are ground-truth aligned pairs, but in the consolidated setting, the dangling entities could bring in erroneous alignment which contaminates the alignment of other pairs.

G.2 Class Prior Estimation Supplementary Experiment (RQ2).

We hope further to verify the estimation and convergence results of iPULE of loss convergence. We list the corresponding loss convergence results in Fig. 7. The losses under different pre-aligned proportions $(0.25, 0.2, 0.15, 0.1)$ on the GA-DBP15K constitute a group of statistical data, and the corresponding loss mean and standard deviation of this set of statistical data are displayed.

On the other hand, the loss difference is a direct indication of convergence in iPULE's implementation. Thus, we plot the histogram figure of the DBP2.0 (w/ -minus & -plus) of the corresponding loss difference for more comprehensive. With the progress of the algorithm, and the statistical number of the difference of the smaller loss function occupied the maximum. This shows the convergence of iPULE in this data set from another aspect.

It is worth noticing that, there are performance fluctuations during the constitution of an ideal embedding space during the cold start stage. The figure plotted covers only the cold start subsequent procedure.

G.3 Embedding Dimension Selection (RQ3).

Although a higher embedding dimension may encode richer information, an overly high dimension leads to performance decline. We select the GNN dimension according to the principle of [22]. Let

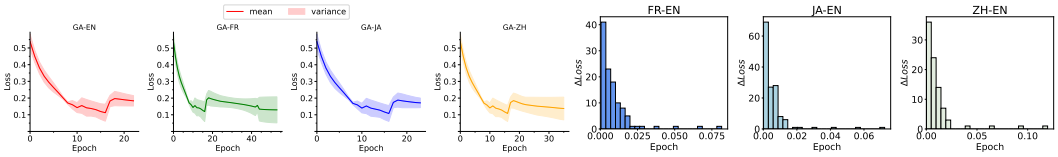

Figure 7: Visualization of loss convergence on DBP2.0 and GA-DBP15K.

the dimension of embedding be d and the number of entities is N. According to the feature entropy in [22], it holds that $d > 8.33 \log N$ by the Johnson-Lindenstrauss lemma [15] that the vector dimension is at $\mathcal{O}(\log N)$ order. In most of our settings, N is approximately 10^5, and thus d is set to 128.

Dimension		ZH-EN			EN-ZH			JA-EN			EN-JA			FR-EN			EN-FR		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Ours	64	.278	.446	.342	.224	.501	.310	.325	.410	.362	.239	.470	.317	.164	.224	.189	.135	.257	.178
	96	**.281**	**.451**	**.346**	**.226**	**.505**	**.312**	.329	.415	.367	.241	.474	.320	.172	.235	.199	.143	.271	.187
	128	.280	.448	.344	.225	.502	.311	**.330**	**.416**	**.368**	**.243**	**.477**	**.322**	**.177**	**.242**	**.204**	**.151**	**.287**	**.198**

Table 9: The entity alignment performance over different embedding dimensions on DBP2.0.

As shown in Tab. 9, due to the varying number of entities in datasets, the embedding dimension at the optimal performance varies. For example, the top performance is achieved on ZH-EN and EN-ZH when the embedding dimension is 96 but is obtained on JA-EN, EN-JA, FR-EN, and EN-FR with an embedding dimension of 128. As a compromise, we simplify the embedding representation of edges to enable FR-EN and EN-FR to run with an embedding dimension of 128 with limited memory (For more details, please check our open source code). A higher alignment performance can be achieved if no compromise is made. As we observe, the optimal performance is typically achieved at the theoretically chosen d. This also indicates our approach has a memory cost at the order of $\mathcal{O}(\log N)$.

G.4 Efficiency (RQ4)

The previous works concerning the dangling problem have not analyzed its efficiency in their experiments. Thus we only report the efficiency of our methods without baseline comparison. We evaluate the efficiency of our work on Tab. 10 including alignment search time as 'Inference Time', KEESA average training time as 'Average Training Time', and GPU memory cost on three different datasets of DBP2.0. Data obtained from these three datasets with the top three node numbers is a robust indicator of the efficiency of our method.

We gathered the mean value of 5 inference time costs for each dataset with the corresponding CPU and GPU memory consumption. Meanwhile, the average training time for each period from early to late is calculated. We enumerate the average training time of epochs 1-20, 21-25, 26-30, 31-35, 36-40, and 41-45.

Cause GPU is employed for not only model training but also inference, as shown on Tab. 10, the inference speed is still very impressive. Specifically, we split the large similarity matrix into multiple independent row blocks to perform the nearest searches within each block, which are well suited for GPU parallel processing.

It's noteworthy that the average training time correspondingly increases as training progresses from early to late stages. More quasi-supervised information incorporated by us accounts for that. To be specific, as the training deepens, we repeatedly conduct preliminary alignment tests while we gather more and more entity pairs mutually closest under a given metric. The entity pairs serve as the pre-aligned anchor nodes, i.e., the quasi-supervisory information mentioned above.

Besides, we list the CPU and GPU memory consumption required for our work. Memory consumption is influenced by various factors such as complex allocation algorithms, model parameter scales, and hyperparameters. In this problem, we put more attention on triples which characterize one KG, revealing an approximate proportionality between the number of triples and memory consumption.

https://doi.org/10.52202/079017-2507

Datasets	Triples	Inference Time	Average Training Time (from 1 to 45 training epochs)						CPU Memory	GPU Memory
			1-20	21-25	26-30	31-35	36-40	41-45		
$\text{DBP2.0}_{\text{ZH-EN}}$	872,935	48.78s	11.21s/it	21.16s/it	25.67s/it	28.17s/it	29.21s/it	30.14s/it	10.8GB	32.5GB
$\text{DBP2.0}_{\text{JA-EN}}$	1,015,545	120.76s	28.14s/it	53.99s/it	63.43s/it	68.27s/it	70.61s/it	72.80s/it	11.9GB	32.6GB
$\text{DBP2.0}_{\text{FR-EN}}$	2,089,909	382.48s	90.18s/it	158.18s/it	190.65s/it	-	-	-	27.7GB	60.2GB

Table 10: Efficiency performance of our work on DBP2.0. The measurement of average training time is 's/it', which indicates seconds per iteration. One iteration here represents one training epoch. '-' indicates the absence of data due to training termination.

G.5 Baseline Comparison Under Different Ratios of Pre-aligned Seeds (RQ5).

Comparing the proposed method with strong baseline models under different ratios of pre-aligned seeds would better demonstrate Lambda's superiority. The experimental baseline includes MtransE w/ BR the SOTA method in previous works, which is also the only open-source method. The results are shown in the Table. 11.

Methods	Ratios	ZH-EN			EN-ZH			JA-EN			EN-JA			FR-EN			EN-FR		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
MtransE w/ BR	10%	.161	.141	.148	.105	.127	.114	.102	.102	.128	.102	.128	.113	.133	.085	.102	.096	.072	.083
	20%	.264	.267	.265	.186	.251	.213	.179	.180	.251	.180	.251	.210	.215	.151	.179	.167	.138	.150
	30%	**.312**	.362	.335	**.241**	.376	.294	.314	.363	.336	**.251**	.358	.295	**.265**	.208	.233	**.231**	.213	.222
Lambda	10%	**.236**	**.346**	**.280**	**.197**	**.385**	**.261**	**.262**	**.315**	**.286**	**.206**	**.360**	**.262**	**.179**	**.230**	**.201**	**.153**	**.260**	**.193**
	20%	.262	**.399**	**.316**	**.215**	**.446**	**.290**	**.300**	**.368**	**.330**	**.226**	**.417**	**.293**	**.217**	**.286**	**.247**	**.182**	**.324**	**.233**
	30%	.279	**.447**	**.344**	.219	**.489**	**.303**	**.324**	**.409**	**.362**	.234	**.460**	**.310**	.234	**.320**	**.271**	.192	**.363**	**.251**

Table 11: Performance of Lambda and MtransE w/ BR under different ratios of pre-aligned of 10%, 20%, and 30%. **Bold** indicates optimal performance.

G.6 LightEA as Strong Baseline for Comparison (RQ6).

LightEA [28] is recommended as a strong baseline for Lambda. We fixed LightEA's code to include dangling entities into the alignment candidates and evaluated its performance on DBP2.0. Hits@1 and Hits@10 are evaluated in a similar way to the dangling-unaware methods in our paper, as listed below. In comparison, Lambda still outperforms LightEA.

Methods	ZH-EN		JA-EN		FR-EN	
	H@1	H@10	H@1	H@10	H@1	H@10
LightEA	60.5%	82.9%	61.4%	**84.1%**	-	-
Lambda	**62.6%**	**84.7%**	**62.1%**	84.0%	**44.1%**	**69.3%**

Table 12: Comparison of Lambda and LightEA under relaxed setting. '-' indicates the absence of data due to out of time.

H Discussion

H.1 Alignment Direction

As we found, the alignment problem with dangling cases has a deeper issue concerning the classification of imbalanced datasets. It originated from the observation that the alignment performance from the source to the target is different from the other direction. The work of [33] has observed that on DBP2.0, choosing the alignment direction from a less populated KG (e.g., ZH, JA, FR) to a more populated KG (e.g., EN) enjoys a higher alignment accuracy but the other way around would lead to a noticeable performance drop. Meanwhile, the dangling entity detection on EN-XX has a higher F1 score than XX-EN, as shown in Tab. 13.

By analysis, we think it may be attributed to an improper indication of the dangling entity detection power on imbalanced datasets. This error in removing the predicted dangling entity would accumulate hurting the alignment task. To verify the point, we introduce a trivial classifier that makes a simple choice to classify all entities as dangling (positive) ones, and the detection results are reported in

Datasets	Dangling Detection						Entity Alignment		
	Our Work			Trivial			Our Work		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
ZH-EN	.763	**.925**	.836	.583	**1**	.736	**.279**	.447	**.344**
EN-ZH	**.844**	.909	**.875**	**.609**	**1**	**.756**	.219	**.489**	.303
JA-EN	.807	**.836**	.821	.580	**1**	.734	**.324**	**.409**	**.362**
EN-JA	**.880**	.809	**.843**	**.605**	**1**	**.753**	.234	.320	.271
FR-EN	.615	**.772**	.685	.439	**1**	.610	**.234**	.320	**.271**
EN-FR	**.732**	.749	**.740**	**.554**	**1**	**.715**	.192	**.363**	.251

Table 13: Dangling entities detection by our classifier v.s. a trivial one on DBP2.0.

Tab. 13. As all unlabeled entities are trivially classified as dangling ones, the detection metrics of the trivial classifier are all falsely high. The more populated source KG usually has more dangling entities (positives) and thus yields a higher precision in detection. Meanwhile, since the detection classifier actually is not working, more dangling entities participate in the alignment phase, resulting in poor alignment performance. This has explained why EN-XX has a higher dangling detection performance but a lower alignment accuracy compared to the other direction.

The root of this issue is that matchable and dangling entities comprise imbalanced categories in the classification task, but the corresponding metric is inappropriate. Hence boosting the detection performance does not necessarily improve the alignment performance. We believe more practical indicators of imbalanced datasets should be introduced to the alignment problem.

H.2 The Similarity between Dual-AMN and Lambda

The differences between the proposed GNN and Dual-AMN include:

Aggregation:

1. The adaptive dangling indicator r_{e_j} is included in Lambda for eliminating dangling pollution.

2. The indicator r_{e_j} is concatenated as a part of the entity feature.

Attention:

1. The attention is scaled by r_{e_j} to filter dangling information.

2. Relation r_k's embedding $\mathbf{h}_{\mathbf{r_k}}$ is linked to the adaptive dangling indicator of the associated entity r_{e_j}, and thus the attention in Eq. (2) models the relationship between the relation and the entity.

https://doi.org/10.52202/079017-2507

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: see **Abstract** and **Introduction. 1**.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explained the trade-off of our method in **How does our method work? 5.3** for a slightly inferior precision reported in Tab. 12 and Tab. 4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: According to the order of appearance, we sort out and give the specific proof in the appendix.

- The problem setting about PU learning as sec. 2.2
- Proof for Lemma 3.
- Proof for Theorem 1.
- Proof for Theorem 2.
- Proof for Theorem 3.
- Proof for Lemma 2.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We introduce the method proposed in this paper in detail in two sections, **Selective Aggregation with Spectral Contrastive Learning** 3 and **Iterative Positive-Unlabeled Learning for Dangling Detection** 4, and use the Alg. 1 to describe the latter in pseudocode. Meanwhile, we gave implementation details at the beginning of the Experiment 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

https://doi.org/10.52202/079017-2507

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the code and data in supplemental material which is described in a documented readme.md file.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We gave the main implementation details at the beginning of the Experiment 5. *Statistics of the experimental dataset and baselines* in appendix F and *additional experiment* in appendix G also cover that including dataset construction details and hyperparameter selection criteria.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provide the corresponding mean and standard deviation curves in Fig. 7 by calculating the loss function of different alignment ratios $0.25, 0.2, 0.15, 0.1$, and the corresponding mean and standard deviation are drawn. Other experimental data have also been measured many times to take the mean value.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We gave GPU and CPU resources needed for the experiment in **Implementation Detail** part at the beginning of the Experiment 5. Additionally, time of execution such as training & inference time is provided in Efficiency. G.4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The dataset construction and usage do not contain any information that endangers personal privacy, and it is licensed.

Guidelines:

https://doi.org/10.52202/079017-2507

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: In this paper, we give all the sufficient reference materials. We provide the code and data in the supplemental material and describe them in a documented readme.md file, where more required information is clarified.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce the dataset GA16K, GA-DBP15K and DBP2.0-minus & -plus in detail in the appendix F. We provide the code and data in the supplemental material and describe them in a documented readme.md file, where more required information is clarified.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

https://doi.org/10.52202/079017-2507

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.