A Training Procedure of Our Approach

The detailed training procedure of our model is described in Algorithm 1.

Our framework takes as input G_a that consists of the two target knowledge graphs (KGs), a set of prior aligned entity pairs. Like a standard neural network training process, we provide a set of tuneable parameters like the number of epochs, N.

As described in Section 4.1 of the main paper, the training of our framework consists of three stages: (1) preliminary entity alignment, (2) approximating relation representations, and (3) joint entity and relation alignment.

In the first stage, we utilize GCNs to learn entity representation, $\mathbf{X}'$, to embed entities of various KGs for preliminary entity alignment. Next, we use the entity embeddings to approximate relation representations (i.e., $\boldsymbol{r}$ at line 7) to align relations across KGs. When the performance of preliminary entity alignment model has become stable, we enter the third stage (lines 12-23). In this stage, we learn a model to try to incorporate the relation representations into entity embeddings ($\boldsymbol{e}_{joint}$ at line 19) to obtain the joint entity representations ($\mathbf{X}_{joint}$ at line 20), and continue using GCNs to iteratively integrate neighboring structural information to achieve better entity and relation representations.

Algorithm 1 Training framework of our model.

Input: Graph G_a, set of aligned entity pairs $\mathbb{L}$, number of epochs N, interval to regenerate negative samples T, number of negative samples $\mathcal{K}$.
Output: Parameters Ω.

```
1: Mode = Preliminary, epoch = 0
2: while Model_has_not_stabilized do
3:     X' = HighwayGCN(G_a, Ω)
4:     if epoch mod T == 0 then
5:         L' = SampleNegative(X', L, K)
6:     end if
7:     r = f(H_r, T_r), ∀r ∈ R_a
8:     L_pre = MarginLoss(X', L, L')
9:     Back propagate errors and update parameters Ω
10:    epoch = epoch + 1
11: end while
12: Mode = Joint
13: while epoch < N do
14:    X' = HighwayGCN(G_a, Ω)
15:    if epoch mod T == 0 then
16:        L' = SampleNegative(X', L, K)
17:    end if
18:    r = f(H_r, T_r), ∀r ∈ R_a
19:    e_joint = g(e, R_e), ∀e ∈ E_a
20:    L_joint = MarginLoss(X_joint, L, L')
21:    Back propagate errors and update parameters Ω
22:    epoch = epoch + 1
23: end while
24: return Ω
```

B Statistical Information of Alignment

Table 1 displays the statistical characteristics of entities and relations which are correctly predicted in both preliminary and joint alignment stage, only in joint alignment stage, and in neither of the stages. From Table 1a, it can be observed that the more neighbor entities and relations an entity has, the more likely and earlier it will be aligned. So the neighbor information is of great importance to entities, and our model utilizes this information effectively. Also, an entity has higher possibility to be aligned in the joint alignment stage if more entities and relations in its neighborhood have been aligned, for these entities and relations offer precise information and help the embedding of the current entity.

Table 1b shows the influence of neighbor entities to relation alignment. There is a gap in the frequency of occurrence in triples between relations that are aligned preliminarily or not, because relations with abundant objects are embedded accurately. Moreover, when a relation fails to be aligned in the preliminary stage, it is more likely to be aligned in the joint stage if it appears frequently, as the embedding of relations will improve with the embedding of entities together.

(a) Entity Alignment

Statics		Pre✓ Joint✓	Pre× Joint✓	Pre× Joint×
#Nbr Ent.	G_1	7.32	6.47	6.25
	G_2	9.52	8.51	8.33
#Nbr Rel.	G_1	3.94	3.73	3.51
	G_2	4.67	4.27	4.30
Pre Aligned Ent. (%)		83	79	75
Pre Aligned Rel. (%)		96	95	95

(b) Relation Alignment

Statics		Pre✓ Joint✓	Pre× Joint✓	Pre× Joint×
Freq.	G_1	85.59	17.83	16.86
	G_2	135.74	48.75	40.93

Table 1: Statistics of entity and relation alignments. Pre✓ Joint✓ indicates the set of entities or relations which are predicted correctly in both preliminary and joint alignment, and so on. #Nbr Ent. and #Nbr Rel. denote the average number of neighbor entities and relations, respectively. Freq. denotes the average frequency of occurrences of relations in each dataset.