A Supplemental Material

A.1 An Implementation of CPL

CPL is a general framework whose components, two agents, are all replaceable. In our experiments, we modify MINERVA (Das et al., 2017) to construct the reasoner, and modify PCNN-ATT (Lin et al., 2016) to construct the fact extractor. Here we briefly introduce a specific implementation of CPL based on PCNN-ATT and MINERVA (see also Fig. 3 for illustration) in details.

Fact Extractor. PCNN-ATT is an effective Relation Extraction approach containing mainly two parts: the sentence encoder and the attention-selector. The sentence encoder encodes each sentence into a vector given the labeled entity pair and their positions in the sentence. We organize the sentences into sentence bags. The sentences in the same bag share the same entity-pair label ($x_\omega = (e^s, e^o)$). For each sentence bag, we modify PCNN-ATT to produces a predictive probability distribution over all relations in the vocabulary: $\Phi(\omega) = F_{pcnn}(x_\omega)$. Suppose at time step t during the inference, the reasoner is at entity e^t. We need to suggest several edges pointing to different entities from e^t to enrich the reasoner's action space. We use PCNN-ATT to make predictions on several sentence bags whose labels all contain e^t and get a distribution set: $[\Phi(\omega_1), ..., \Phi(\omega_k)] = F_{pcnn}([x_{\omega_1}, ..., x_{\omega_K}]), w_k = (e^t, r^t_k, e^t_k), k \in [1, K]$. The distribution set can be seen as a score set over different edges. We can define a stochastic policy based on the scores by sampling the edges according to the scores. In the original PCNN-ATT setting, the score indicates the confidence of linking e^s and e_o w.r.t. the respective predicted relation. According to the previous reward definition, we can construct the policy distributions over all candidate edges based on these output scores via softmax and provide the extractor with the most relevant edges.

Graph Reasoner. To extend MINERVA as our graph reasoner, we adopt random action drop-out (random dropping KG-edges) to unite the KG-edges and corpus-extracted edges into a joint action space of fixed size. Specifically, for a query triple (e_s, r_q, e_q), it predicts e_q through finding a path from e_s to e_q w.r.t. r_q. At time step t, the observed state s^t is (e^s, r_q, e^t, h^t) as defined before. The history information before t is a sequence of edges, which is encoded into a vector with LSTM : $h^t = LSTM(h_{t-1}, [r^t; e^t])$. The reasoner should select one edge from the joint action space defined above w.r.t. e^t. The MINERVA model F_{mine} takes in the state embedding and output the softmax scores w.r.t. each action (out-edge). Then we adopt adaptive sampling (discussed below) to select the action to proceed.

Training pipeline. A formal training algorithm is given in Algorithm 1.

Algorithm 1 CPL(G, C, b_r, b_e, p_l, e_a, e_m)

```
Require: Knowledge graph $G$, corpus $C$, # of
    batches training the reasoner $b_r$, # of batches
    training the extractor $b_e$, hyper parameters for
    learning $p_l$, # of epochs applying adaptive
    sampling $e_a$, maximal epochs $e_m$.
Ensure: CPL model
Initialize the reasoner and extractor.
Register Adam optimizer with $p_l$.
for $e$ = 0: $e_m$ do
    for 0 to max-batches do
        if $e < e_a$ then
            Generate training sequences with
            adaptive sampling.
        elseGenerate training sequences with
            normal sampling.
        end if
        Store the training sequences into
        replay memories.
        Sample from the replay memory to
        train the reasoner for $b_r$ batches.
        Sample from the replay memory to
        train the reasoner for $b_r$ batches.
    end for
end for
```

A.2 Training techniques

we will introduce a few techniques we use to increase training efficiency.

The lack of positive training samples is a common challenge for most RL algorithms. We use two techniques to accumulate positive experiences for the agents, *model pre-training* and *adaptive sampling*.

i) Model Pre-training. To get proper initialization, we pre-train the fact extractor and reasoner. In this way, at the beginning of the joint training, we can expect the agents to generate plausible experiences immediately.

ii) Adaptive Sampling. The policy learned by the agents can be regarded as a distribution of choosing certain actions given the states. Usually, we sample the actions multiple times according to the distribution to generate multiple experiences. In the pre-training stage, the reasoner is unaware of the facts in the texts. It tends to ignore the new facts suggested by the extractor. To facilitate interactions between two agents and encourage exploration, we reconstruct the distribution to ensure the extracted edges to be chosen with higher probability. Specifically, at time step t, the action space of the reasoner is the union of KG-edges and extracted edges, i.e., $A^t = \{(r, e)|(e^t, r, e) \in KG\} \cup \{(r', e')|(e^t, r', e') \in \text{corpus } C\}$. The reasoner will score all the actions in A^t, and we increase the scores of extracted edges adaptively so that they have higher priority over the KG-edges. Whereas we cannot keep this priority all the time, it twists the true data or pattern distribution. Hence after a number of iterations, we stop the adaptive sampling and use the immediate policy distribution for sampling.

To increase exploration efficiency, the fact extractor samples multiple edges given its learned policy to add to the reasoner's joint action space (Fig. 3). We collect the experiences with above techniques and store them into two replay memories (Mnih et al., 2013) for two agents separately.

A.3 Experiment Details

A.3.1 Datasets and Codes[5]

We study the datasets and find that the relation distributions of the two datasets are very imbalanced. There are not enough reasoning paths for some relation types. Moreover, some relations are meaningless and of no reasoning value. We select a subset of the relations for each dataset as the reasoning tasks. There are enough reasoning paths for the path-based models to learn on these relations. They are also pretty informative and widely concerned according to the opinions of the domain experts we interviewed. The details are in Table 2. Specifically, we first divide the dataset into train, validation, and test sets in the proportion $8:1:1$ randomly. Then we only keep the triples of the concerned relations in the validation and test set.

A.3.2 Training Setup

We list the parameter and experimental set-ups for all the algorithms in this section. The parameters not mentioned below have minor influences on the performance, so we follow the default configurations in their codes.

ComplEx, DistMult, &TransE We use the implementation from OpenKE [6]. We set the embedding dimension as 100. We train each model for 600 iterations and 800 samples within each iteration.

RotatE We use the implementation supplied by its author [7]. We set the embedding dimension as 100 (although the recommended value is 1000; we set this to avoid biases and training hurdles). We train each model for a total of 150k steps, with batch size of 256.

ConvE We use the public code for evaluation[8]. We set the embedding dimension as 200, training for 50 epochs. We use the same negative sampling ratio (i.e., 1:1) as what we use in the above OpenKE models for FB60K; and 1:all negative sampling for UMLS.

Rc-net We use the code provided by the authors of paper (Xu et al., 2014). We use all the default parameters except that we set the sample number as 48. In this way, we ensure that the training sample quantity used in Rc-net is the same as others.

JointNRE We get the code from the authors of paper (Han et al., 2018). We use the PCNN-ATT as the sentence encoder and transE as the KG embedding method, which is the best-performing combination according to the authors. PCNN is trained for 20 iterations, during which the KG is trained by selecting 100 samples each batch, reaching 7,500 iterations at the end of training. Since the pre-trained word vector is a 50-dimension set, the embedding dimension is also 50.

LINE+TransE To train the word network and entity dictionary, the window is set to 5. For the embedding part, embedding dimension is set to 50; 100 samples are selected in each epoch, while the number of epochs is stable at 1,000,000.

MINERVA We use the code [9] for evaluation. Since our Joint model approach requires MINERVA as a base model, we use the same em-

[5] The two datasets aforementioned in this paper and data pre-processing codes are in the supplementary materials and also available at `https://drive.google.com/file/d/1hCyPBjywpMuShRJCPKRjc7n2vHpxfetg/view?usp=sharing`. The codes in the supplemental material is our implementation of the CPL.

[6] https://github.com/thunlp/OpenKE

[7] https://github.com/DeepGraphLearning/KnowledgeGraphEmbedding

[8] https://github.com/TimDettmers/ConvE

[9] https://github.com/shehzaadzd/MINERVA

bedding sizes and hidden sizes on the MINERVA training and our model training. To get our result we trained it for 400 iterations at a batch size of 64 samples on FB60K. We set the iteration-number to 1000 and batch-size to 64 for UMLS.

To better reflect the models' capabilities, all models related to MINERVA are added reverse edge triples. Considering the inevitable fluctuations of this reinforcement learning model, we use three random keys 55, 83 and 5583 to initiate training and reach an average result for the three runs.

Our model In total we train 400 iterations for FB60K (considering time factors) and 1000 for UMLS; For first 200 iterations, we use BFS to search positive paths with higher priority on PCNN-ATT suggested edges. In each BFS iteration, 100 samples are selected. The learning rate is set to 0.001, and the batch size is 64, the same as MINERVA.

A.4 Case Study

1. Two-step is the naive solution to OKGR. For the two-step model, we filter the corpus-edges with the output scores (in [0,1]) of PCNN-ATT. 0 means adding all the edges to the KG, while 1 means adding nothing. We find the best threshold (producing the best reasoning model) for UMLS-PubMed is 0.5 and 0 for FB60K-NYT10. Two-step adds about 85,000 edges to UMLS and 90,000 to FB60K under the corresponding thresholds, whereas CPL adds about 8,000 edges to UMLS and 1,500 for FB60K.

The two-step model performance is inferior to CPL and MINERVA on all the datasets (Table 3, 4). The reasons are that 1) most of the extracted edges use in the two-step model are noises; 2) adding so many edges significantly enlarges the explore space for reasoning. Selecting the correct out-edge at each step becomes more difficult. Lack sufficient positive experiences, with same iterations, the two-step model cannot learn the underlying patterns well.

3. Figure 7 shows the inference cases randomly sampled from the FB60K-NYT10 dataset. We select three relations and randomly sample several query cases from the test set. We track down the inference paths for each query case and mark the edges suggested by the extractor. Further, we track back to the raw text data to pick out the sentences from which the extractor extract the relevant facts. For example, for the query triple (gorgonzola, /location/location/contains_inv, m.0bzty), the concerned relation is "/location/location/contains_inv". A possible pattern to infer the relation is "/location/location/contains_inv" ∧ "/location/location/contains" ∧ "/location/location/contains_inv" → "/location/location/contains_inv". The specific path found by the reasoner is (gorgonzola, /location/location/contains_inv, Italy) → (Italy, /location/location/contains, san_siro) → (san_siro, /location/location/contains_inv, m.0bzty). Among them, edge (Italy, /location/location/contains, san_siro) is a new edge suggested by the extractor, which is extracted from the sentence "the san_siro is one of 25 stadiums in italy that the country s security and sports officials condemned for not having in place certain security measures aimed at cutting down on fan violence ".

Relation	Path pattern	Query triple case	Inference path	Edge suggested by PCNN	Related text
/location/location/contains_inv	/location/location/contains_inv → /location/location/contains→ /location/location/contains_inv	[m.0491k9k]-[indiana]: */location/location/contains_inv*	[m.0491k9k]-[united_states_of_america]: */location/location/contains_inv* [united_states_of_america]-[fort_wayne]: */location/location/contains* [fort_wayne]-[indiana]: */location/location/contains_inv*	[fort_wayne]-[indiana]: */location/location/contains_inv*	as many as 300 people originally from darfur are living in ***fort_wayne*** , with others scattered across smaller ***indiana*** cities like elkhart , south bend and goshen .
			[m.0491k9k]-[united_states_of_america]: */location/location/contains_inv* [united_states_of_america]-[taylor_university]: */location/location/contains* [taylor_university]-[indiana]: */location/location/contains_inv*	[taylor_university]-[indiana]: */location/location/contains_inv*	beers , who is 47 , knows the subculture well: his father was a dean at ***taylor_university ,*** a christian college in ***indiana*** that still forbids most dancing on campus .
			[m.0491k9k]-[united_states_of_america]: */location/location/contains_inv* [united_states_of_america]-[midwest]: */location/location/contains* [midwest]-[indiana]: */location/location/contains_inv*	[midwest]-[indiana]: */location/location/contains_inv*	southwestern michigan the stretch of coastline along michigan 's southwestern border attracts a steady stream of traffic from chicago , as well as from ***indiana*** , southern michigan , missouri and ohio , though agents say they are also seeing more buyers from outside the ***midwest*** .
		[gorgonzola]-[m.0bzty]: */location/location/contains_inv*	[gorgonzola]-[italy]: */location/location/contains_inv* [italy]-[san_siro]: */location/location/contains* [san_siro]-[m.0bzty]: */location/location/contains_inv*	[italy]-[san_siro]: */location/location/contains*	the ***san_siro*** is one of 25 stadiums in ***italy*** that the country 's security and sports officials condemned for not having in place certain security measures aimed at cutting down on fan violence .
/people/person/nationality	/people/person/place_of_birth→ /people/person/place_lived_inv → /people/person/nationality	[m.026qxz]-[united_states_of_america]: */people/person/nationality*	[m.026qxz]-[hollywood]: */people/person/place_of_birth* [hollywood]-[mischa_barton]: */people/person/place_lived_inv* [mischa_barton]-[united_states_of_america]: */people/person/nationality*	[hollywood]-[mischa_barton]: */people/person/place_lived_inv*	her tone , dulcet when she talks about the women she has dressed-- ***hollywood*** nymphets like keira knightley and ***mischa_barton***-- turns raspy when she catalogs her woes .
			[m.026qxz]-[hollywood]: */people/person/place_of_birth* [hollywood]-[elvis_presley]: */people/person/place_lived_inv* [elvis_presley]-[united_states_of_america]: */people/person/nationality*	[hollywood]-[elvis_presley]: */people/person/place_lived_inv*	-lrb- ***hollywood*** has taken notice he has a small role as ***elvis_presley*** in the coming biopic spoof '' walk hard , '' for which judd apatow is a writer .-rrb-
			[m.026qxz]-[hollywood]: */people/person/place_of_birth* [hollywood]-[blake_edwards]: */people/person/place_lived_inv* [blake_edwards]-[united_states_of_america]: */people/person/nationality*	[hollywood]-[blake_edwards]: */people/person/place_lived_inv*	his ***hollywood*** connections included writing two screenplays with ***blake_edwards*** , '' the man who loved women '' and '' that 's life ! '
		[nellie_mckay]-[united_states_of_america]: */people/person/nationality*	[nellie_mckay]-[new_york_city]: */people/place_lived/location_inv* [new_york_city]-[dr._john]: */people/person/place_lived_inv* [dr._john]-[united_states_of_america]: */people/person/nationality*	[new_york_city]-[dr._john]: */people/person/place_lived_inv*	in his later years , pomus became an elder statesman in the ***new_york_city*** songwriter set , a larger-than-life connection to a lost era , knocking around town with ***dr._john*** and lou reed before succumbing to lung cancer in 1991 .
/location/neighborhood/neighborhood_of	/location/location/containedby→ /location/location/contains→ /location/neighborhood/neighborhood_of	[east_hampton]-[new_york_city] : */location/neighborhood/neighborhood_of*	[east_hampton]-[united_states_of_america]: */location/location/containedby* [united_states_of_america]-[st._george]: */location/location/contains* [st._george]-[new_york_city]: */location/neighborhood/neighborhood_of*	[st._george]-[new_york_city]: */location/neighborhood/neighborhood_of*	the ***st._george*** neighborhood near the ferry to manhattan is the closest thing to a downtown district in the borough , but it lacks the vibrancy of other sections of ***new_york_city*** that have become havens for young professionals and artists , said jonathan bowles , who wrote the study for the center for an urban future , a public policy group .
		[port_chester]-[new_york_city]: */location/neighborhood/neighborhood_of*	[port_chester]-[united_states_of_america]: */location/location/containedby* [united_states_of_america]-[st._george]: */location/location/contains* [st._george]-[new_york_city]: */location/neighborhood/neighborhood_of*		
		[westborough]-[new_york_city]: */location/neighborhood/neighborhood_of*	[westborough]-[united_states_of_america]: */location/location/containedby* [united_states_of_america]-[st._george]: */location/location/contains* [st._george]-[new_york_city]: */location/neighborhood/neighborhood_of*		

Figure 7: **A Case study on discovered paths on FB60K-NYT10.**We randomly pick three relations and show how CPL performs reasoning based on the KG and text corpus. Red texts are the relations. $[xxx]$-$[xxx]$ represents [subject entity]-[object entity]. The bold italic words in the sentences means where we extract the relations.

Model / Dataset	**20%**			**50%**			**100%**		
	$\sigma(Hits@5)$	$\sigma(Hits@10)$	$\sigma(MRR)$	$\sigma(Hits@5)$	$\sigma(Hits@10)$	$\sigma(MRR)$	$\sigma(Hits@5)$	$\sigma(Hits@10)$	$\sigma(MRR)$
TransE (Bordes et al., 2011)	0.0013	0.0011	0.0017	0.0015	0.0014	0.0014	0.0015	0.0015	0.0011
DisMult (Yang et al., 2014)	0.0015	0.0015	0.0017	0.0010	0.0015	0.0017	0.0017	0.0020	0.0012
ComplEx (Trouillon et al., 2016)	0.0040	0.0030	0.0030	0.0013	0.0025	0.0029	0.0024	0.0026	0.0027
ConvE (Dettmers et al., 2018)	0.0031	0.0043	0.0026	0.0020	0.0027	0.0019	0.0032	0.0026	0.0027
RC-Net (Xu et al., 2014)	0.0009	0.0016	0.0016	0.0013	0.0016	0.0015	0.0018	0.0007	0.0017
TransE+Line	0.0015	0.0013	0.0008	0.0014	0.0005	0.0013	0.0013	0.0012	0.0014
JointNRE (Han et al., 2018)	0.0015	0.0012	0.0016	0.0015	0.0016	0.0007	0.0016	0.0012	0.0015
MINERVA (Das et al., 2017)	0.0100	0.0118	0.0148	0.0856	0.1009	0.0550	0.0974	0.0849	0.1253
Two-Step	0.0140	0.0137	0.0095	0.0290	0.0279	0.0309	0.0343	0.0368	0.0614
CPL (our method)	0.0028	0.0017	0.0044	0.0131	0.0033	0.0547	0.0040	0.0010	0.0227

Table 5: **Performance variance of KG reasoning on the FB60K-NYT10 dataset.** Reinforcement learning methods do suffer from variances between different runs.

Model / Dataset	**20%**		**40%**		**70%**		**100%**	
	$\sigma(Hits@5)$	$\sigma(Hits@10)$	$\sigma(Hits@5)$	$\sigma(Hits@10)$	$\sigma(Hits@5)$	$\sigma(Hits@10)$	$\sigma(Hits@5)$	$\sigma(Hits@10)$
TransE (Bordes et al., 2011)	0.0027	0.0023	0.0035	0.0039	0.0034	0.0032	0.0029	0.0020
DisMult (Yang et al., 2014)	0.0020	0.0034	0.0031	0.0017	0.0029	0.0032	0.0033	0.0022
ComplEx (Trouillon et al., 2016)	0.0026	0.0022	0.0035	0.0029	0.0036	0.0007	0.0016	0.0024
ConvE (Dettmers et al., 2018)	0.0028	0.0027	0.0030	0.0028	0.0025	0.0033	0.0029	0.0020
RC-Net (Xu et al., 2014)	0.0024	0.0030	0.0013	0.0030	0.0030	0.0014	0.0026	0.0022
TransE+Line	0.0027	0.0040	0.0026	0.0029	0.0024	0.0013	0.0036	0.0013
JointNRE (Han et al., 2018)	0.0008	0.0016	0.0026	0.0028	0.0028	0.0034	0.0028	0.0019
MINERVA (Das et al., 2017)	0.0171	0.0195	0.0327	0.0217	0.0565	0.0499	0.0575	0.0678
Two-Step	0.0072	0.0088	0.0193	0.0178	0.0217	0.0025	0.0021	0.0094
CPL (our method)	0.0155	0.0020	0.0090	0.0031	0.0166	0.0028	0.0155	0.0033

Table 6: **Performance variance of KG reasoning on the UMLS-PubMed dataset.**