Exploring Effective Inter-Encoder Semantic Interaction for Document-Level Relation Extraction

Liang Zhang[1,2], **Zijun Min**[1,2], **Jinsong Su**[1,2], **Pei Yu**[1,2], **Ante Wang**[1,2], **Yidong Chen**[1,2]
[1]School of Informatics, Xiamen University, China
[2]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China
lzhang@stu.xmu.edu.cn, {jssu,ydchen}@xmu.edu.cn

Abstract

In document-level relation extraction (RE), the models are required to correctly predict implicit relations in documents via relational reasoning. To this end, many graph-based methods have been proposed for this task. Despite their success, these methods still suffer from several drawbacks: 1) their interaction between document encoder and graph encoder is usually unidirectional and insufficient; 2) their graph encoders often fail to capture the global context of nodes in document graph. In this paper, we propose a document-level RE model with a Graph-Transformer Network (GTN). The GTN includes two core sublayers: 1) *the graph-attention sublayer* that simultaneously models global and local contexts of nodes in the document graph; 2) *the cross-attention sublayer*, enabling GTN to capture the non-entity clue information from the document encoder. Furthermore, we introduce two auxiliary training tasks to enhance the bidirectional semantic interaction between the document encoder and GTN: 1) *the graph node reconstruction* that can effectively train our cross-attention sublayer to enhance the semantic transition from the document encoder to GTN; 2) *the structure-aware adversarial knowledge distillation*, by which we can effectively transfer the structural information of GTN to the document encoder. Experimental results on four benchmark datasets prove the effectiveness of our model. Our source code is available at https://github.com/DeepLearnXMU/DocRE-BSI.

1 Introduction

Relation extraction (RE) is an important task in the community of information extraction (IE), which aims to identify the relations between entities in a given text. While most previous studies focused on extracting relational triples from a single sentence [Zeng *et al.*, 2015; Zhang *et al.*, 2018; Baldini Soares *et al.*, 2019], i.e., sentence-level RE, many researchers have recently begun to explore RE at the document level [Zeng *et al.*, 2020; Zhou *et al.*, 2021; Jiang *et al.*, 2022]. Unlike sentence-level RE, document-level RE aims to extract all relation triples from the input document. It is usually more challenging since documents normally contain a large number of implicit relations that can only be identified with the help of relational reasoning. According to the statistics in [Yao *et al.*, 2019], in the commonly-used DocRED dataset, about 61.6% of relational facts can only be correctly predicted with the assistance of relational reasoning.

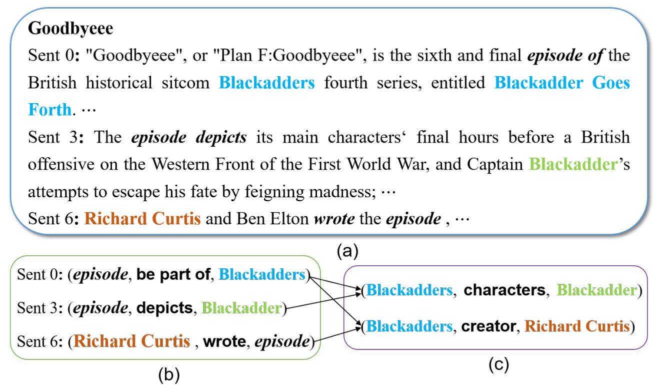

Figure 1: (a) shows an input document, where different colors represent different entities and *Italics* indicate non-entity clue words, which provide useful information for predicting the inter-sentence relations in (c). (b) illustrates the relations between the non-entity word *episode* and three entities, which are directly expressed by the three sentences. By using *episode* as pivots, the two complex inter-sentence relations in (c) can be easily inferred from the relations in (b) (See the arrows from (b) to (c)).

Due to the advantages of graph neural networks (GNNs) in relational reasoning, graph-based methods are widely adopted in document-level RE [Zeng *et al.*, 2020; Xu *et al.*, 2021b; Peng *et al.*, 2022]. These methods first use a pre-trained language model (PLM) as the encoder to obtain the document's contextual representation, and then leverage dependency structures, heuristics, or structured attention to construct a document graph [Peng *et al.*, 2017; Christopoulou *et al.*, 2019; Nan *et al.*, 2020]. Finally, GNNs are applied to encode the document graph for relational reasoning.

Although graph-based methods have achieved competitive performance in document-level RE [Zeng *et al.*, 2020; Peng *et al.*, 2022], they still suffer from several drawbacks. **First**, in these methods, the interaction between document encoder (PLM) and graph encoder (GNN) is normally insufficient and unidirectional, severely limiting the performance of the model. Specifically, when encoding the document graph, these methods typically only consider entities while ignoring non-entity words that might provide crucial clues for relational reasoning. This suggests that the semantic transition from the document encoder to the graph encoder (**PLM**→**GNN**) is insufficient. As illustrated in Figure 1, since the entities "*Blackadders*" and "*Richard Curtis*" appear in different sentences, it is usually difficult to correctly predict their relation only using their own information. Note that the non-entity word "*episode*" respectively co-occurs with these two entities in a different sentence, and their relations *(episode, be part of, Blackadders)* and *(Richard Curtis, wrote, episode)* can be easily identified. With these two relations, we can further infer the "*creator*" relation between "*Blackadders*" and "*Richard Curtis*" (See Figure 1(c)). Besides, these methods do not directly pass the structural information of the graph encoder to the document encoder (**GNN**→**PLM**), resulting in that the document encoder cannot directly benefit from the graph encoder [Xu *et al.*, 2021a]. **Second**, during document graph encoding, these methods usually update the node representations by only aggregating the information of their neighbor nodes. However, this approach only focuses on capturing the local context of a considered node while neglecting its global context [Hu *et al.*, 2019; Wan *et al.*, 2021; Wang *et al.*, 2021], which significantly reduces the reasoning ability of the model.

To deal with the above issues, we propose a graph-based document-level RE model with the bidirectional semantic interaction between the document encoder and the graph encoder. As shown in Figure 2, we first use a **PLM encoder** to encode the input document. Then, on the top of the encoder, we construct a heterogeneous document graph **(HDG)** that consists of three types of nodes, namely mention node, entity node, and document node, and three types of edges, i.e., intra-sentence edge, intra-entity edge, and document edge. Lastly, we propose a new graph encoder, Graph-Transformer Network **(GTN)**, to encode HDG and generate more expressive entity representations. Particularly, our GTN includes two core sublayers: the graph-attention sublayer and the cross-attention sublayer. The former is a multi-head self-attention variant with four attention heads, where the first three attention heads are applied to capture the local context of a considered node from its neighbors, and the last one is used to capture the global context of the node from all other nodes. Furthermore, via the cross-attention sublayer, GTN can capture the non-entity clue information from PLM encoder to enhance the reasoning ability of the model.

To effectively enhance the bidirectional semantic interaction between the PLM encoder and our GTN, we introduce two auxiliary tasks into our model training: the graph node reconstruction and the Structure-aware Adversarial Knowledge Distillation **(SA-KD)**. To implement graph node reconstruction, we first mask the feature vectors of some nodes in HDG, and then train GTN to reconstruct the original features of these nodes. In this way, we can effectively train GTN to obtain more clue information from PLM encoder via the cross-attention sublayer. Besides, we employ SA-KD

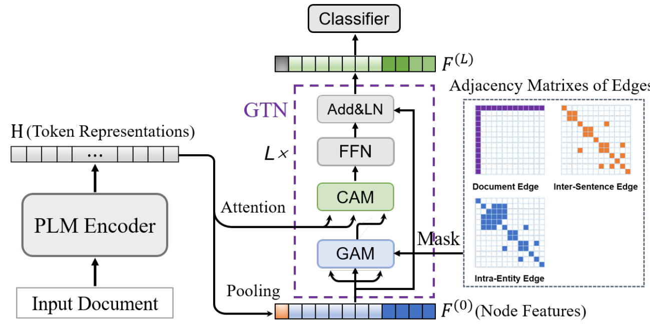

Figure 2: The overall architecture of our model. First, we encode the input document via PLM encoder to obtain word-level contextual representations H. Then, we heuristically construct an HDG composed of three kinds of nodes and three types of edges. Finally, we encode HDG using GTN to obtain more expressive entity representations, which are fed into the classifier for relation prediction.

to guide the PLM encoding using the structural information of GTN. Specifically, we develop a discriminator to distinguish the node representations generated by PLM encoder and GTN. Meanwhile, we regard PLM encoder as the generator (student), which is trained to produce node representations conforming to the distributions of GTN (teacher), so that the discriminator cannot distinguish. By alternately optimizing the discriminator and the generator, our model can effectively transfer the structural information of GTN to PLM encoder. Note that our discriminator is more tolerant than predefined distance functions in conventional knowledge distillation, such as cosine distance, because it does not require PLM encoder and GTN to output the exact same node representation. By doing so, we can effectively prevent the model training from collapsing.

To demonstrate the effectiveness and generality of our model, we conduct comprehensive experiments on four public datasets, of which results show that our model consistently outperforms all competitive baselines.

2 Methodology

In this section, we describe in detail our model and its training. As illustrated in Figure 2, our model consists of two components: the **PLM encoder** and the **Graph-Transformer Network (GTN)**. First, the input document is encoded with PLM encoder to obtain contextual representations of tokens (Section 2.1). Then, on the basis of these representations, we construct a Heterogeneous Document Graph (**HDG**) and encode it using GTN to produce more expressive entity representations (Section 2.2). Finally, we give a detailed description of our model training (Section 2.3).

2.1 The PLM Encoder

Following [Zhou *et al.*, 2021; Tan *et al.*, 2022a], we first use the special token "$*$" to mark the start and end of mentions in the input document D. Then, we encode D using PLM encoder to obtain contextual representations $\mathbf{H} \in R^{|D|\times d}$ of tokens, where d is the dimension of PLM encoder hidden states. Finally, we take the contextual representation of the token $[CLS]$ as the document's contextual representation h_D, and the contextual representation of "*" at the start position of the mention m_j as its contextual representation $h(m_j)$. Particularly, we merge all the mention representations of the entity e_i via *logsumexp pooling* [Jia *et al.*, 2019] to generate its global contextual representation $h(e_i) = \log \sum_{j=1}^{N_{e_i}} \exp(h(m_j^i))$, where N_{e_i} refers to the mention number of e_i.

2.2 The Graph-Transformer Network

Graph Representation

To model dependencies among entities or mentions, we first heuristically construct an HDG. Our HDG includes three types of nodes: mention node, entity node, and document node. We initialize the feature vectors of these nodes using the contextual representations of mention, entity, and document obtained from PLM encoder, respectively. Meanwhile, we introduce three types of edges into HDG:

- **Intra-Entity Edge**. We introduce intra-entity edges to fully connect mention and entity nodes of the same entity. This allows us to efficiently aggregate different mention representations of entities to generate better entity representations.
- **Intra-Sentense Edge**. The intra-sentence edges are utilized to fully connect mention and entity nodes that cooccur in the same sentence. In this way, we can effectively model the interaction among different entities.
- **Document Edge**. All mention and entity nodes are connected to the document node via document edges. By using this type of edge as pivots, we can effectively improve the semantic interaction between distant entities.

Graph Encoding

Then, we encode the HDG with the GTN to generate more expressive entity representations. To facilitate the calculation of GTN, we combine the feature vectors of all nodes into a feature matrix $F^{(0)} \in R^{N \times d}$, where N represents the number of nodes. Concurrently, we build an adjacency matrix $E_k \in R^{N \times N}$ ($k \in \{1, 2, 3\}$) for each type of edge.

As shown in Figure 2, our GTN contains L identical layers, each of which consists of four sublayers: the graph-attention sublayer, the cross-attention sublayer, the feed-forward neural network sublayer, and the layer normalization sublayer. Next, we detail the two core sublayers of GTN, i.e., the first two sublayers.

Graph-Attention Sublayer. This sublayer is a variation of multi-head self-attention with four attention heads, where the first three heads are used to capture the local contexts of nodes in HDG, and the fourth head is used to capture the global contexts of nodes in HDG. With the help of this sublayer, GTN can simultaneously model the local and global contexts of nodes in HDG.

We use the first three attention heads to model the three types of edges in HDG, respectively. Specifically, for each type of edge, we utilize its adjacency matrix E_i as the attention mask matrix in the corresponding attention head. Formally, at the $(l+1)$-th layer, the i-th attention head is calculated as follows:

$$
\begin{aligned}
F_i^{(l+1)} &= A(F^{(l)} W_i^V), \\
A &= \mathrm{softmax}\Big(\frac{(F^{(l)} W_i^Q)(F^{(l)} W_i^K)^\mathsf{T}}{\sqrt{d}} - \big((1-E_i) \circ \mathrm{Inf}\big)\Big),
\end{aligned} \tag{1}
$$

where W_i^K, W_i^Q, and W_i^V are trainable parameters, Inf refers to infinity. Obviously, by introducing Inf, each node is limited to only focus on its neighbors in HDG.

Specially, in the fourth attention head, we do not perform the mask operation, which allows each node to pay attention to all other nodes. To prevent losing structural information of HDG in this head, we add the entity embedding emb_e and the sentence embedding emb_s to each node:

$$
\begin{aligned}
F_4^{(l+1)} &= \mathrm{softmax}\Big(\frac{(\widetilde{F}^{(l)} W_4^Q)(\widetilde{F}^{(l)} W_4^K)^\mathsf{T}}{\sqrt{d}}\Big)(F^{(l)} W_4^V), \\
\widetilde{F}^{(l)} &= F^{(l)} + \mathrm{emb}_e + \mathrm{emb}_s,
\end{aligned} \tag{2}
$$

where W_4^K, W_4^Q, and W_4^V are parameter matrixes.

Cross-Attention Sublayer. Through this sublayer, we expect that GTN can extract clue information from PLM encoder to improve the model's reasoning abilities. To adapt to long documents and capture more diverse clue information, we develop two types of attention heads in the cross-attention sublayer: global attention head and local attention head. In global attention head, the nodes in HDG can consider all the words in the document to capture global clue information. In local attention head, via the mask operation, each node can only focus on the words in the sentence where it is located, so as to capture local clue information.

Based on the final output $F^{(L)}$ of GTN, we utilize a bilinear classifier to predict the relations of entity pairs:

$$
\begin{aligned}
p_{s,o} &= \sigma(\mathbf{z}_s^\mathsf{T} W_r \mathbf{z}_o), \\
\text{where} \ \ \mathbf{z}_s &= \tanh(W_s[F^{(L)}[e_s], c_{s,o}]), \\
\mathbf{z}_o &= \tanh(W_o[F^{(L)}[e_o], c_{s,o}]).
\end{aligned} \tag{3}
$$

where W_r, W_s, and W_o are trainable parameters, $F^{(L)}[e_s]$ and $F^{(L)}[e_o]$ denote the feature vectors of entities e_s and e_o, respectively, and $c_{s,o}$ represents the localized context embedding [Zhou *et al.*, 2021] utilized to enhance the representation of entity pair (e_s, e_o). More specifically, $c_{s,o}$ is computed as

$$
c_{s,o} = \mathbf{H}^\mathsf{T} \frac{A_s \circ A_o}{\mathbf{1}^\mathsf{T}(A_s \circ A_o)}, \tag{4}
$$

where A_s and A_o denote the PLM last-layer attention weights of entities e_s and e_o to all tokens in the document, respectively, and $\circ$ refers to element-wise multiplication.

2.3 Model Training

To effectively enhance the bidirectional semantic interaction between PLM encoder and GTN, we introduce two auxiliary tasks into our model training: the graph node reconstruction and the Structure-aware Adversarial Knowledge Distillation (SA-KD). Thus, the final training objective of our model contains three loss items: the relation classification loss $\mathcal{L}_{\mathcal{R}}$, the graph node reconstruction loss $\mathcal{L}_{\mathcal{N}}$, and the SA-KD loss $\mathcal{L}_{\mathcal{A}}$:

$$
\mathcal{L} = \mathcal{L}_{\mathcal{R}} + \alpha \mathcal{L}_{\mathcal{N}} + \beta \mathcal{L}_{\mathcal{A}}. \tag{5}
$$

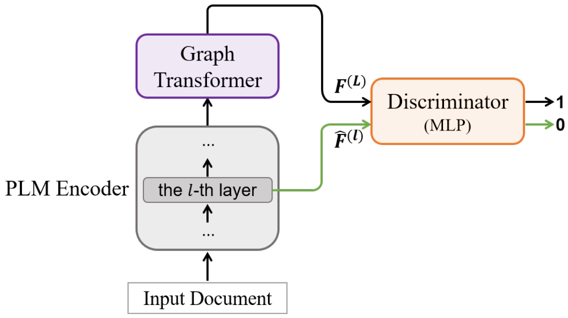

Figure 3: Illustration of our SA-KD.

where α and β are hyper-parameters, which are empirically set to 0.1 and 0.01, respectively.

Relation Classification Loss $\mathcal{L}_{\mathcal{R}}$. To alleviate the imbalance relation distribution issue in document-level RE, we adopt the adaptive thresholding loss [Zhou *et al.*, 2021] as our relation classification loss. Specifically, we introduce a special relation class TH and use its logits $\mathrm{logit}_{\mathrm{TH}}$ as the adaptive threshold value for each entity pair to distinguish between positive relations $\mathcal{P}_{\mathcal{T}}$ and negative relations $\mathcal{N}_{\mathcal{T}}$:

$$\begin{aligned}\mathcal{L}_{\mathcal{R}}=-\Bigg(\sum_{r\in\mathcal{P}_{\mathcal{T}}}\log\Big(\frac{\exp(\mathrm{logit}_r)}{\sum_{r'\in\mathcal{P}_{\mathcal{T}}\cup\{\mathrm{TH}\}}\exp(\mathrm{logit}_{r'})}\Big)\Bigg)\\ -\log\Big(\frac{\exp(\mathrm{logit}_{\mathrm{TH}})}{\sum_{r'\in\mathcal{N}_{\mathcal{T}}\cup\{\mathrm{TH}\}}\exp(\mathrm{logit}_{r'})}\Big).\end{aligned} \tag{6}$$

Node Reconstruction Loss $\mathcal{L}_{\mathcal{N}}$. To effectively improve the semantic transition from PLM encoder to GTN, we introduce the graph node reconstruction task into our model training. Specifically, we randomly sample and mask some nodes in HDG, and then train GTN to reconstruct the initial features of these nodes. To recover the original features of masked nodes, GTN has to leverage its cross-attention sublayer to capture more clue information from PLM encoder. Formally, we define the node reconstruction loss as follows:

$$\mathcal{L}_{\mathcal{N}}=\frac{1}{|\mathcal{V}_{mask}|}\sum_{v\in\mathcal{V}_{mask}}\Big(1-\mathrm{Cosine}\big(F^{(L)}[v],F^{(0)}[v]\big)\Big) \tag{7}$$

Where $\mathcal{V}_{mask}$ represents the masked nodes.

SA-KD Loss $\mathcal{L}_{\mathcal{A}}$. The purpose of the SA-KD is to transfer the structural information in GTN to PLM encoder. As shown in Figure 3, we first construct a node feature matrix $\widehat{F}^{(l)}$ from the l-th layer output of PLM encoder. Then, we develop an MLP discriminator $\mathcal{C}$ to distinguish whether the considered node is from the output $\widehat{F}^{(l)}$ of PLM encoder or the output $F^{(L)}$ of GTN. Finally, we use binary cross entropy loss as $\mathcal{L}_{\mathcal{A}}$:

$$\mathcal{L}_{\mathcal{A}}=\log P(1|\mathcal{C}(F^{(L)})+\log P(0|\mathcal{C}(\widehat{F}^{(l)}) \tag{8}$$

where $\mathcal{C}(\cdot)$=Sigmoid(MLP($\cdot$)). Meanwhile, we regard PLM encoder as the generator and train it to produce node representations conforming to the distributions of GTN, so that the discriminator cannot distinguish. In this way, we can guide PLM encoder to learn more expressive entity representations using the structural information of GTN. Notably, we simultaneously distill the structural information in GTN to multiple intermediate layers $\{l\}$ of PLM encoder using multiple distinct discriminators. Specifically, we empirically set $\{l\}$ to $\{6,12\}$ in the BERT encoder and $\{12,18,24\}$ in the RoBERTa-large encoder.

Finally, during model training, PLM encoder and GTN are trained to minimize $\mathcal{L}$, while the discriminator is trained to maximize $\mathcal{L}_{\mathcal{A}}$.

3 Experiments

3.1 Datasets and Evaluation Metrics

We evaluate our model on four commonly-used datasets:

- **DocRED** [Yao *et al.*, 2019] is a large-scale document-level RE dataset with 96 predefined relations, which is constructed from Wikipedia and Wikidata. It contains 5,053 documents, which is divided into 3,053 documents for training, 1,000 for development, and 1,000 for test. Since DocRED contains a considerable number of false-negative samples, we also conduct experiments on its two revised versions, i.e., **Revisit-DocRED** [Huang *et al.*, 2022] and **Re-DocRED** [Tan *et al.*, 2022b].
- **DWIE** [Zaporojets *et al.*, 2021] is an entity-centric multi-task dataset containing 602 documents for training, 98 for development, and 99 for test. In this dataset, there are about 26% of entity pairs expressing more than one relation of the predefined 62 target relations. We followed [Ru *et al.*, 2021] to preprocess the DWIE dataset.

Following previous studies [Yao *et al.*, 2019; Ru *et al.*, 2021], we utilize micro F_1 and micro Ign F_1 as our evaluation measures. Ign F_1 denotes the F_1 score excluding the relational facts that are shared by the training and development/test sets.

3.2 Settings

Our model is developed based on Huggingface's Transformers [Wolf *et al.*, 2020] and PyTorch. We use cased BERT-base [Devlin *et al.*, 2019] or RoBERTa-large [Liu *et al.*, 2019] as our encoder. To optimize our model, we use AdamW [Loshchilov and Hutter, 2019] as our optimizer, which is equipped with a weight decay of 1e-4 and a linear warmup [Goyal *et al.*, 2017] for the first 6% training steps. All hyper-parameters are tuned on the development set.

3.3 Baseline Models

We compare our model with the existing Transformer-based and Graph-based models.

- **Transformer-based models** directly employ PLMs to learn better entity representations for document-level RE, including BERT-TS [Wang *et al.*, 2019], HIN-BERT [Tang *et al.*, 2020], CorefBERT [Ye *et al.*, 2020], and ATLOP-BERT [Zhou *et al.*, 2021].
- **Graph-based models** leverage GNNs to enhance the reasoning ability of document-level RE models, including EoG [Christopoulou *et al.*, 2019], DHG [Zhang *et al.*, 2020], GEDA [Li *et al.*, 2020], LSR [Nan *et al.*,

Model	Re-DocRED				DocRED			
	Dev		Test		Dev		Test	
	Ign F_1	F_1	Ign F_1	F_1	Ign F_1	F_1	Ign F_1	F_1
GEDA-BERT [Li *et al.*, 2020]	–	–	–	–	54.52	56.16	53.71	55.74
LSR-BERT [Nan *et al.*, 2020]	–	–	–	–	52.43	59.00	56.97	59.05
GLRE-BERT [Wang *et al.*, 2020]	–	–	–	–	–	–	55.40	57.40
GAIN-BERT [Zeng *et al.*, 2020]	71.99*	73.49*	71.88*	73.44*	59.14	61.22	59.00	61.24
HeterGSAN-BERT [Xu *et al.*, 2021b]	–	–	–	–	58.13	60.18	57.12	59.45
SSAN-BERT [Xu *et al.*, 2021a]	–	–	–	–	56.68	58.95	56.06	58.41
BERT-TS [Wang *et al.*, 2019]	–	–	–	–	–	54.42	–	53.92
HIN-BERT [Tang *et al.*, 2020]	–	–	–	–	54.29	56.31	53.70	55.60
CorefBERT [Ye *et al.*, 2020]	–	–	–	–	55.32	57.51	54.54	56.96
ATLOP-BERT [Zhou *et al.*, 2021]	73.35*	74.22*	73.22*	74.02*	59.22	61.09	59.31	61.30
SIRE-BERT [Zeng *et al.*, 2021]	–	–	–	–	59.82	61.60	60.18	62.05
DocuNet-BERT [Zhang *et al.*, 2021]	73.68†	74.65†	73.60†	74.49†	59.86	61.83	59.93	61.86
KD-DocRE-BERT [Tan *et al.*, 2022a]	73.76†	74.69†	73.67†	74.55†	60.08	62.03	60.04	62.08
KMGRE-BERT [Jiang *et al.*, 2022]	73.33*	74.44*	73.39*	74.46*	–	–	–	–
Ours-BERT	**75.03**	**75.85**	**74.85**	**75.77**	**60.86±0.20**	**62.73±0.17**	**60.77**	**62.75**

Table 1: Experimental results on the development and test sets of Re-DocRED and DocRED. We report the mean and standard deviation on the development set by conducting five experiments with different random seeds. Besides, we report the test scores of the best checkpoint on the development set. * indicates that scores are reported in [Jiang *et al.*, 2022]. Results with † are obtained by our reproduction.

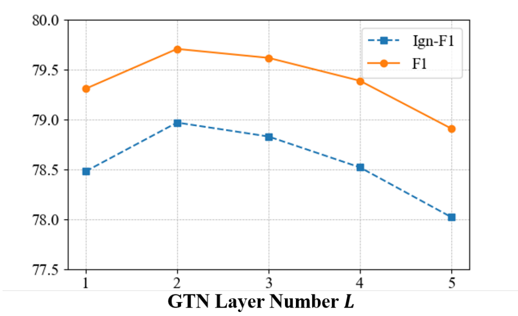

Figure 4: The performance of our model with different GTN Layer Number L on the development set of Re-DocRED.

2020], GLRE [Wang *et al.*, 2020], GAIN [Zeng *et al.*, 2020], HeterGSAN [Xu *et al.*, 2021b], SIRE [Zeng *et al.*, 2021], and SSAN [Xu *et al.*, 2021a].

In addition, we select several recent competitive models, including DocuNet [Zhang *et al.*, 2021], KD-DocRE [Tan *et al.*, 2022a], and KMGRE [Jiang *et al.*, 2022] for comparison.

3.4 Effect of GTN Layer Number L

To illustrate the influence of the hyper-parameter L on our model, we report the performance of our model with different GTN layer numbers in Figure 4. Like previous graph-based methods [Zeng *et al.*, 2020; Peng *et al.*, 2022], our model achieves the best performance when L is set to 2. Meanwhile, we also note that the performance of our model is not so sensitive to L. Finally, we set L=2 in all subsequent experiments.

3.5 Main Results

Results on RE-DocRED and DocRED. As illustrated in Table 1, our model consistently outperforms all baselines on RE-DocRED and DocRED datasets. Moreover, we draw several interesting conclusions:

First, compared with the improvements on DocRED, our model achieves greater gains on RE-DocRED that contains more relational facts involving relational reasoning. It suggests that our model indeed performs better in reasoning scenarios.

Second, compared with the graph-based SOTA model, GAIN-BERT, our model obtains improvements of **2.33** F_1 and **1.51** F_1 points on the test sets of RE-DocRED and DocRED. These results demonstrate that our model can better capture the dependencies among entities and mentions to improve the reasoning ability of the model.

Third, our model also surpasses recent SOTA models, including DocuNet-BERT and KD-DocRE-BERT, which leverage the dependencies among entity pairs to enhance their reasoning abilities. This fully illustrates again the excellent reasoning ability of our model.

Results on Revisit-DocRED. Unlike DocRED and RE-DocRED, the training set of Revisit-DocRED contains a large number of false-negative samples, but its test set does not. As illustrated in Table 2, our model consistently and significantly outperforms all competitive baseline models on this datasets, demonstrating that our model is robust to noisy data.

Results on DWIE. To confirm the generalizability of our model, we also conduct experiments on the DWIE dataset. From Table 2, we find that our model significantly outperforms KD-DocRE-BERT by **1.63** Ign F_1 and **1.54** F_1 points on the test sets of DWIE, achieving new SOTA performance on this dataset.

3.6 Ablation Study

To further comprehend the contributions of different components on our model, we conduct an ablation study by remov-

Model	Revisit-DocRED		DWIE			
	Test		Dev		Test	
	Ign F_1	F_1	Ign F_1	F_1	Ign F_1	F_1
GAIN-BERT [Zeng *et al.*, 2020]*	41.27	41.64	58.63	62.55	62.37	67.57
SSAN-BERT [Xu *et al.*, 2021a]*	41.64	41.92	58.62	64.49	62.58	69.39
ATLOP-BERT [Zhou *et al.*, 2021]*	41.62	41.90	63.57	69.96	67.56	74.36
DocuNet-BERT [Zhang *et al.*, 2021]†	42.94	43.29	65.65	71.52	70.04	76.79
KD-DocRE-BERT [Tan *et al.*, 2022a]†	43.22	43.68	65.84	71.78	70.27	77.01
KMGRE-BERT [Jiang *et al.*, 2022]*	42.78	43.16	65.56	71.40	69.94	76.71
Ours-BERT	**44.84±0.16**	**45.33±0.11**	**67.42±0.37**	**73.45±0.31**	**71.90**	**78.55**

Table 2: Performance on the development/test set of Revisit-DocRED and DWIE. Here, we use the same experimental settings as in Table 1. * indicates that scores are reported in [Jiang *et al.*, 2022]. Results with † are obtained by our reproduction.

Model	Ign F_1	F_1
Ours-BERT	**75.03**	**75.85**
SA-KD→KD	73.68	74.39
w/o Node reconstruction task	74.31	75.17
w/o Local head in cross-attention	74.58	75.42
w/o Global head in cross-attention	74.40	75.26
w/o Cross-attention	74.19	75.01
w/o The fourth head in graph-attention	74.21	75.04
Graph-attention→Self-attention	73.01	73.78

Table 3: Ablation study of our model on the dev set of Re-DocRED.

Model	Intra-F_1	Inter-F_1
LSR-BERT [Nan *et al.*, 2020]	65.26	52.05
GAIN-BERT [Zeng *et al.*, 2020]	67.10	53.90
BERT-TS [Wang *et al.*, 2019]	61.90	47.28
SIRE-BERT [Zeng *et al.*, 2021]	68.07	54.01
Ours-BERT	**68.49**	**55.74**

Table 4: Intra-F_1 and Inter-F_1 scores on the dev set of DocRED.

ing different components from our model. Specifically, we compare our model with the following variants in Table 3.

(1) *SA-KD→KD*. In this variant, we replace our SA-KD with the conventional KD that employs cosine similarity as the predefined distance function to measure the gap between nodes generated by PLM encoder and GTN. From **Line 3** in Table 3, we observe that this replacement results in a significant performance drop. The possible explanation is that the predefined distance functions in the conventional KD cause the collapse of model training during knowledge distillation, whereas our SA-KD can effectively avoid this collapse issue.

(2) *w/o Node reconstruction task*. When node reconstruction task is removed from our model training, the performance of our model drops by 0.72 Ign F_1 and 0.68 F_1 points (See **Line 4** in Table 3). This result demonstrates that this task can effectively train our cross-attention sublayer to extract more clue information from PLM encoder.

(3) *w/o Local head in cross-attention* and *w/o Global head in cross-attention*. To capture diverse clue information and adapt to long documents, we equip the cross-attention sublayer with two types of attention heads: local and global attention heads. To demonstrate this, in these two variants, we respectively remove the local and global attention heads from our cross-attention sublayer, both of which negatively impact the performance of our model (See **Line 5-6** in Table 3).

(4) *w/o Cross-attention*. In this variant, we exclude the cross-attention sublayer from our model, which results in a performance decrease in our model (See **Line 7** in Table 3). This suggests that our cross-attention sublayer can effectively capture non-entity clue information from PLM encoder to improve the reasoning ability of the model.

(5) *w/o The fourth head in graph-attention*. To investigate the effectiveness of the global context of nodes in HDG, we remove the fourth attention head from our graph-attention sublayer. As shown in **Line 8** of Table 3, this variant causes a significant performance decline, which confirms the contribution of node global context on the model performance.

(6) *Graph-attention→Self-attention*. In this variant, we replace our graph-attention sublayer with a standard multi-head self-attention sublayer, where each attention head acts on a fully connected graph. This change leads to a significant performance drop of 2.02 Ign F_1 and 2.07 F_1 points (See **Line 9** in Table 3). For this result, we speculate that the standard multi-head self-attention sublayer loses the structural information of HGD and causes the over-smoothing problem.

3.7 Analysis of Reasoning Performance

To further illustrate the reasoning ability of our model, following [Nan *et al.*, 2020; Zeng *et al.*, 2020], we also report Intra-F_1, Inter-F_1 and Infer-F_1 scores. When calculating Intra-F_1 and Inter-F_1, we solely consider intra-sentence and inter-sentence relations, respectively. Meanwhile, we calculate the Infer-F_1 score using the test files supplied by Zeng *et al.*, [2020]. This metric aims to assess the ability of the model in multi-hop reasoning.

As shown in Table 4, our model outperforms all baseline models on Intra-F_1 and Inter-F_1 metrics. We notice that our model achieves more significant improvements on Inter-F_1 than on Intra-F_1, demonstrating that our model is excellent at extracting inter-sentence relations. In addition, extracting inter-sentence relations is usually more challenging than intra-sentence ones, so that Inter-F_1 can more effectively reflect the reasoning ability of the model.

Model	Infer-F_1	P	R
BERT-RE [Zeng *et al.*, 2020]	39.62	34.12	47.23
GAIN-BERT [Zeng *et al.*, 2020]	46.89	38.71	59.45
Ours-BERT	**49.92**	**42.82**	**59.87**
w/o Cross-attention	49.36	42.06	59.74
w/o SA-KD	48.22	40.49	59.61
Graph-attention→Self-attention	46.29	38.02	59.15

Table 5: Infer-F_1 scores on the development set of DocRED.

From Table 5, in term of Infer-F_1 metrics, we observe that our model also obtains a significant improvement compared to all baseline models. Specifically, our model yields an improvement of **3.03** Infer-F_1 points over GAIN-BERT that is the graph-based SOTA model. Meanwhile, removing either cross-attention sublayer or SA-KD from our model causes a significant decline in our model's performance (See **Line 7-8** in Table 5). Furthermore, when we replace our graph-attention sublayer with a standard multi-head self-attention sublayer, our model performance sharply drops by **3.63** Infer-F_1 points (See **Line 9** in Table 5). These results demonstrate that each component in our model can enhance the reasoning ability of the model.

4 Related Work

Recently, document-level RE has attracted an increasing amount of interest. The dominant methods for document-level RE can be roughly divided into Transformer-based methods and graph-based methods.

Transformer-based Methods. Since PLMs have achieved striking success in natural language processing (NLP), some researchers directly employ Transformer-based PLMs for document-level RE, which focus on extracting more useful information from PLM to enhance the representations of entities [Wang *et al.*, 2019; Tang *et al.*, 2020; Zhou *et al.*, 2021; Zhang *et al.*, 2022; Zhang *et al.*, 2023]. For example, Wang *et al.*, [2019] propose a two-step process for document-level RE. They first identify whether entity pairs are related, and then predict their relations. Zhou *et al.*, [2021] introduce two techniques, i.e., adaptive thresholding loss and localized context pooling, to alleviate the class imbalance issue and enhance the representations of entity pairs, respectively. However, these methods do not explicitly model the dependencies among entities, which limit the reasoning ability of the model.

Graph-based Methods. In recent years, GNNs have been widely used in various NLP tasks, such as machine translation [Song *et al.*, 2020; Yin *et al.*, 2020b] and sentence ranking [Yin *et al.*, 2019; Yin *et al.*, 2020a; Lai *et al.*, 2021]. To effectively model dependencies among mentions or entities, many researchers also introduce GNNs into document-level RE [Christopoulou *et al.*, 2019; Nan *et al.*, 2020; Zeng *et al.*, 2020]. These methods first construct a document graph with heuristics or dependency information, and use entities or mentions as its nodes. Then, they encode this document graph using GNNs to obtain more expressive entity representations. For example, Nan *et al.*, [2020] propose a latent structure refinement model, which dynamically induces the latent graph structure to facilitate the relational reasoning across sentences. Zeng *et al.*, [2020] construct two graphs of different granularity, i.e., mention-level graph and entity-level graph, to model the interactions among mentions and entities, respectively. Meanwhile, to capture the global context of nodes in document graph, Xu *et al.*, [2021b] and Peng *et al.*, [2022] heuristically incorporate some inference paths into the document graph to enhance the interaction among distant related entities. However, these heuristics are generally incomplete and have poor generalizability. Furthermore, to improve the encoder with the structural information of the document graph, Xu *et al.*, [2021a] incorporate the graph structure into the self-attention of PLM encoder. Nevertheless, this approach introduces many new parameters into PLM encoder, which makes it require a large amount of external data for model training. Notably, graph-based methods usually only consider entity information during relational reasoning while ignoring many non-entity clue information in document, which hinders the further improvement of the model's reasoning ability.

Our work falls into the category of graph-based methods. Specifically, our model consists of PLM encoder and GTN that contains two core components, i.e., graph-attention sublayer and cross-attention sublayer. Through the first sublayer, GTN can simultaneously model the local and global contexts of nodes in the document graph. Meanwhile, we introduce a graph node reconstruction training task, which can effectively train GTN to capture more clue information from PLM encoder via the cross-attention sublayer. Furthermore, inspired by recent studies on knowledge distillation [Chung *et al.*, 2020; He *et al.*, 2022; Zhuang *et al.*, 2022], we propose a SA-KD training task. With this task, we can guide PLM encoder with the structural information of GTN to learn more expressive entity representations. Unlike the conventional adversarial knowledge distillation [Chung *et al.*, 2020; He *et al.*, 2022], we use our GTN as the teacher model and the PLM encoder as the student model, allowing GTN and PLM encoder to promote each other during the distillation process.

5 Conclusion and Future Work

In this paper, we propose a document-level RE model consisting of a PLM encoder and a GTN. Particularly, GTN contains two core components: 1) the graph-attention sublayer that simultaneously models global and local contexts of nodes in HDG; 2) the cross-attention sublayer, which enables GTN to capture the non-entity clue information from PLM encoder. Moreover, to enhance the bidirectional semantic interaction between PLM encoder and GTN, we introduce two auxiliary tasks into model training: 1) the graph node reconstruction that can effectively train our cross-attention sublayer to enhance the semantic transition from PLM encoder to GTN; 2) the SA-KD, by which we can effectively transfer the structural information of GTN to PLM encoder. Experimental results on four commonly-used datasets illustrate that our model outperforms all existing competitive baselines.

In future, we plan to apply our model to other graph-based tasks, such as knowledge graph completion and graph node classification, so as to verify its generality.

Acknowledgments

The authors would like to thank the three anonymous reviewers for their comments on this paper. This research was supported in part by the National Natural Science Foundation of China under Grant Nos. 62076211, U1908216, 62276219, and 61573294.

Contribution Statement

Of all the authors, Liang Zhang and Zijun Min make equal contributions and share co-first authorship. Meanwhile, Jinsong Su and Yidong Chen are the corresponding authors of this paper.

References

[Baldini Soares *et al.*, 2019] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *ACL*, 2019.

[Christopoulou *et al.*, 2019] Fenia Christopoulou, Makoto Miwa, Sophia Ananiadou, and Sophia Ananiadou. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *EMNLP*, 2019.

[Chung *et al.*, 2020] Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. Feature-map-level online adversarial knowledge distillation. In *ICML*, 2020.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[Goyal *et al.*, 2017] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. In *arXiv preprint arXiv:1706.02677*, 2017.

[He *et al.*, 2022] Huarui He, Jie Wang, Zhanqiu Zhang, and Feng Wu. Compressing deep graph neural networks via adversarial knowledge distillation. In *KDD*, 2022.

[Hu *et al.*, 2019] Fenyu Hu, Yanqiao Zhu, Shu Wu, Liang Wang, and Tieniu Tan. Hierarchical graph convolutional networks for semi-supervised node classification. In *IJCAI*, 2019.

[Huang *et al.*, 2022] Quzhe Huang, Shibo Hao, Yuan Ye, Shengqi Zhu, Yansong Feng, and Dongyan Zhao. Does recommend-revise produce reliable annotations? an analysis on missing instances in docred. In *ACL*, 2022.

[Jia *et al.*, 2019] Robin Jia, Cliff Wong, and Hoifung Poon. Document-level n-ary relation extraction with multiscale representation learning. In *NAACL*, 2019.

[Jiang *et al.*, 2022] Feng Jiang, Jianwei Niu, Shasha Mo, and Shengda Fan. Key mention pairs guided document-level relation extraction. In *COLING*, 2022.

[Lai *et al.*, 2021] Shaopeng Lai, Ante Wang, Fandong Meng, Jie Zhou, Yubin Ge, Jiali Zeng, Junfeng Yao, Degen Huang, and Jinsong Su. Improving graph-based sentence ordering with iteratively predicted pairwise orderings. In *EMNLP*, 2021.

[Li *et al.*, 2020] Bo Li, Wei Ye, Zhonghao Sheng, Rui Xie, Xiangyu Xi, and Shikun Zhang. Graph enhanced dual attention network for document-level relation extraction. In *COLING*, 2020.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. In *arXiv preprint arXiv:1907.11692*, 2019.

[Loshchilov and Hutter, 2019] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[Nan *et al.*, 2020] Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. Reasoning with latent structure refinement for document-level relation extraction. In *ACL*, 2020.

[Peng *et al.*, 2017] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *Trans. Assoc. Comput. Linguistics*, 2017.

[Peng *et al.*, 2022] Xingyu Peng, Chong Zhang, and Ke Xu. Document-level relation extraction via subgraph reasoning. In *IJCAI*, 2022.

[Ru *et al.*, 2021] Dongyu Ru, Changzhi Sun, Jiangtao Feng, Lin Qiu, Hao Zhou, Weinan Zhang, Yong Yu, and Lei Li. Learning logic rules for document-level relation extraction. In *EMNLP*, 2021.

[Song *et al.*, 2020] Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. Structural information preserving for graph-to-text generation. In *ACL*, 2020.

[Tan *et al.*, 2022a] Qingyu Tan, Ruidan He, Lidong Bing, and Hwee Tou Ng. Document-level relation extraction with adaptive focal loss and knowledge distillation. In *ACL Findings*, 2022.

[Tan *et al.*, 2022b] Qingyu Tan, Lu Xu, Lidong Bing, and Hwee Tou Ng. Revisiting docred–addressing the overlooked false negative problem in relation extraction. In *EMNLP*, 2022.

[Tang *et al.*, 2020] Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shi Wang, and Pengfei Yin. Hin: Hierarchical inference network for document-level relation extraction. In *KDD*, 2020.

[Wan *et al.*, 2021] Sheng Wan, Shirui Pan, Jian Yang, and Chen Gong. Contrastive and generative graph convolutional networks for graph-based semi-supervised learning. In *AAAI*, 2021.

[Wang *et al.*, 2019] Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William Wang. Fine-tune bert for docred with two-step process. In *arXiv preprint arXiv:1909.11898*, 2019.

[Wang *et al.*, 2020] Difeng Wang, Wei Hu, Ermei Cao, and Weijian Sun. Global-to-local neural networks for document-level relation extraction. In *EMNLP*, 2020.

[Wang *et al.*, 2021] Guangtao Wang, Rex Ying, Jing Huang, and Jure Leskovec. Multi-hop attention graph neural network. In *IJCAI*, 2021.

[Wolf *et al.*, 2020] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *EMNLP*, 2020.

[Xu *et al.*, 2021a] Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *AAAI*, 2021.

[Xu *et al.*, 2021b] Wang Xu, Kehai Chen, Tiejun Zhao, and Tiejun Zhao. Document-level relation extraction with reconstruction. In *AAAI*, 2021.

[Yao *et al.*, 2019] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A large-scale document-level relation extraction dataset. In *ACL*, 2019.

[Ye *et al.*, 2020] Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. Coreferential Reasoning Learning for Language Representation. In *EMNLP*, 2020.

[Yin *et al.*, 2019] Yongjing Yin, Linfeng Song, Jinsong Su, Jiali Zeng, Chulun Zhou, and Jiebo Luo. Graph-based neural sentence ordering. In *IJCAI*, 2019.

[Yin *et al.*, 2020a] Yongjing Yin, Shaopeng Lai, Linfeng Song, Chulun Zhou, Xianpei Han, Junfeng Yao, and Jinsong Su. An external knowledge enhanced graph-based neural network for sentence ordering. *Journal of Artificial Intelligence Research*, 2020.

[Yin *et al.*, 2020b] Yongjing Yin, Fandong Meng, Jinsong Su, Chulun Zhou, Zhengyuan Yang, Jie Zhou, and Jiebo Luo. Novel graph-based multi-modal fusion encoder for neural machine translation. In *ACL*, 2020.

[Zaporojets *et al.*, 2021] Klim Zaporojets, Johannes Deleu, Chris Develder, and Thomas Demeester. Dwie: An entity-centric dataset for multi-task document-level information extraction. *Information Processing & Management*, 2021.

[Zeng *et al.*, 2015] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*, 2015.

[Zeng *et al.*, 2020] Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. Double graph based reasoning for document-level relation extraction. In *EMNLP*, 2020.

[Zeng *et al.*, 2021] Shuang Zeng, Yuting Wu, Baobao Chang, and Baobao Chang. SIRE: Separate intra- and inter-sentential reasoning for document-level relation extraction. In *ACL Findings*, 2021.

[Zhang *et al.*, 2018] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*, 2018.

[Zhang *et al.*, 2020] Zhenyu Zhang, Bowen Yu, Xiaobo Shu, Tingwen Liu, Hengzhu Tang, Wang Yubin, and Li Guo. Document-level relation extraction with dual-tier heterogeneous graph. In *COLING*, 2020.

[Zhang *et al.*, 2021] Ningyu Zhang, Xiang Chen, Xin Xie, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, Luo Si, and Huajun Chen. Document-level relation extraction as semantic segmentation. In *IJCAI*, 2021.

[Zhang *et al.*, 2022] Liang Zhang, Jinsong Su, Yidong Chen, Zhongjian Miao, Min Zijun, Qingguo Hu, and Xiaodong Shi. Towards better document-level relation extraction via iterative inference. In *EMNLP*, 2022.

[Zhang *et al.*, 2023] Liang Zhang, Jinsong Su, Min Zijun, Zhongjian Miao, Qingguo Hu, Biao Fu, Xiaodong Shi, and Yidong Chen. Exploring self-distillation based relational reasoning training for document-level relation extraction. In *AAAI*, 2023.

[Zhou *et al.*, 2021] Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. Document-level relation extraction with adaptive thresholding and localized context pooling. In *AAAI*, 2021.

[Zhuang *et al.*, 2022] Yuanxin Zhuang, Lingjuan Lyu, Chuan Shi, Carl Yang, and Lichao Sun. Data-free adversarial knowledge distillation for graph neural networks. In *IJCAI*, 2022.