SEG-DOC: A SIMPLE YET EFFICIENT GRAPH NEURAL NETWORK FRAMEWORK FOR DOCUMENT UNDERSTANDING

Anonymous ICME submission

ABSTRACT

In recent years, Graph Neural Networks (GNNs) have gained significant attention as an effective approach in the field of document understanding. However, many existing research ignore the complexity of the document structure during constructing graphs and cannot well handle the problem of different neighbor weights during message passing. To address these issues, we propose SEG-Doc: a simple yet efficient graph neural networks framework for document understanding. Firstly, we introduce XY probability graph construction module (XYPG), which can generate graph structures that reflect real-world documents. Secondly, we use document layout information to design the layout attention mechanism (LAM), which is integrated into GNNs to learn richer node representations. We evaluated the effectiveness of SEG-Doc on three benchmark datasets: FUNSD, SROIE and RVL-CDIP. Comprehensive experiments show that the proposed SEG-Doc achieves state-of-the-art performance than competitive baselines on several public downstream benchmarks.

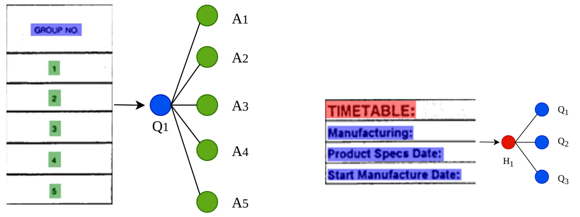
***Index Terms*—** Document AI, Graph Neural Networks, Multimodal, Key Information Extraction, Document Layout Analysis

1. INTRODUCTION

In recent years, deep learning and natural language processing technologies have revolutionized automated document processing. Document understanding, extraction and analysis are fundamental tasks that facilitate information retrieval, decision making and automation in various domains. Visually-rich Document Understanding (VrDU) task not only relies on text information, but also relies on visual and layout information. Therefore, to address VrDU task, it is necessary to effectively fuse the multi-modal information and take advantage of the cross-modal nature of documents. In a multi-modal framework, textual, visual and layout information should be jointly modeled and learned [1–4]. This also provides new research directions and challenges for document understanding.

In the field of document understanding, Key Information Extraction (KIE) and Document Layout Analysis (DLA) are widely concerned subtasks. Most of advanced practices and methods in recent years are based on pre-training transformer large models [5–10]. These models use massive partially labeled data to capture visual, textual and layout cues of documents through pre-training tasks and semi-supervised learning, but bring high computational costs in terms of computing resources and training time. Another mainstream method is GNNs-based [11–16], which regards the document sample as a graph structure composed of segments, and uses the graph neural network model to learn the relationship between the segments. It does not rely on large-scale pre-training, but finding the optimal edge set to create a graph is a very difficult step.

Some GNNs-based methods in the field have limitations in the process of graph construction. Doc2Graph [17] designs a full connection graph, which not only puts a lot of pressure on the model, but also cannot reflect the real-world document structure. CGCN [18] considers connecting edges between adjacent words in four cardinal directions (up, down, left or right), but it does not take into account the complexity of the form. Because form contains a large number of tables and other structures, which will generate many 1-to-n key-value relationships, as shown in Fig. 1. Meanwhile, popular graph neural networks such as GraphSAGE [19] and GCN [20] usually regard all neighbor nodes as having the same weight during message passing. However, in document understanding, different text fragments may have different contributions in meaning and context.



(a) 1 question to 5 answers (b) 1 header to 3 questions.

Fig. 1. A large number of 1-to-n key-value relationships in the form. Red, blue, and green represent Header, Question and Answer labels respectively. Best viewed in color and zoomed-in.

To solve the above issues, in this paper, we propose the SEG-Doc framework. Firstly, documents are constructed as XY probability graphs and multi-modal features are incorporated into nodes and edges. Then, layout attention mechanism (LAM) can be integrated into GNNs to learn richer node representations. Finally, we make predictions on nodes and edges to complete the KIE and DLA tasks. We selected three publicly available benchmark datasets to evaluate the performance of our model, which are the FUNSD [21] dataset for form understanding, the SROIE [22] dataset for receipt understanding and the RVL-CDIP [23] dataset for invoice understanding. Experimental results show that we have achieved good results comparable to strong baselines on three datasets.

In summary, the contributions of this paper are as follows:

- We propose a XY probability graph construction method (XYPG), which can generate graph representations that are more consistent with the structure of real-world documents.
- We design a layout attention mechanism (LAM), which assigns different weights to different neighbors of a node. It can be integrated into GNNs to learn richer node representations.
- We propose SEG-Doc, a GNNs-based framework for document understanding. We mainly focus on key information extraction and document layout analysis tasks in this paper. We evaluate on three benchmark datasets and achieve excellent results.

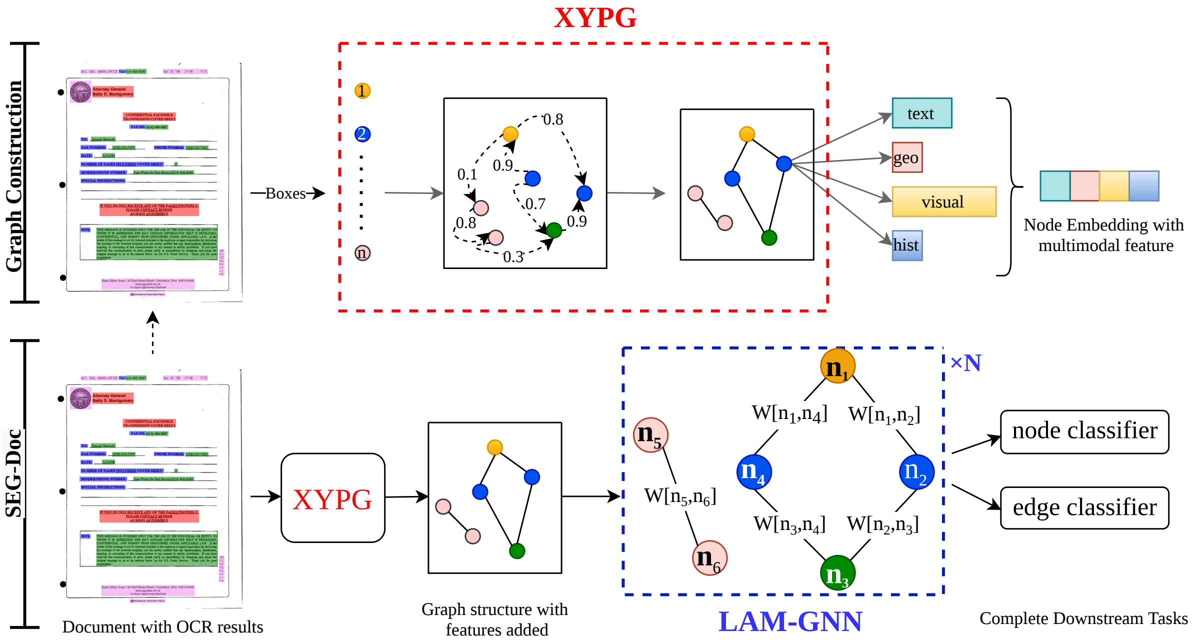


Fig. 2. Overview of SEG-Doc. It contains the two main modules XYPG and LAM-GNN we designed. LAM-GNN represents graph neural network with layout attention mechanism. N represents the number of network layers.

2. RELATED WORK

Key information extraction (KIE) refers to the technology of extracting entities and their relationships from a large amount of unstructured content in documents. Recent research have proven the significance of jointly modeling multi-modal features in KIE performance. Document Layout Analysis (DLA) aims to detect the document structure and the boundaries of each layout region. Table detection and structure recognition received particular attention in recent years. Then, the detected layout segments are classified into predefined document element classes to understand document. In this section, we introduce the excellent works of graph neural networks on KIE and DLA.

Graph Neural Networks For KIE. FUDGE [24] constructs graphs by detecting text lines and calculating edge scores, and uses GCN to predict graph changes. GraphDoc [25] combines GAT [26] and pretraining in a multi-modal framework. It learns generic representations from a collection of unlabeled documents via a self-supervised approach. Doc2Graph [17] improves GraphSAGE by filtering out farther nodes based on their distance during the message passing.

Graph Neural Networks For DLA. Doc-GCN [15] captures the syntactic, semantic, density and appearance features of document layout components by constructing heterogeneous graphs, thereby promoting the model conduct more comprehensive learning of the documents properties. Riba [23] considers introducing graph edge information into the node convolutions and designs a language independent model to accomplish table detection in invoice.

3. METHOD

In this section, we describe our method SEG-Doc in detail. The overall architecture is shown in Fig. 2, which mainly consists of two modules: XYPG module and LAM-GNN module.

Given a scanned document image I, its OCR result contains n bounding boxes. Through XYPG module we can transform the document into a graph structure $G = (V, E, A)$, which represents a set of nodes $V = (v_1, v_2 ... v_n)$ and a set of edges E. In the edge set E, an edge $e_{ij} = (v_i, v_j)$ connects the source node $v_i \in V$ and the target node $v_j \in V$. $A \in R^{n \times n}$ represents the adjacency matrix. We fuse multi-modal features to each node v_i in the node set V to enrich it with semantic, visual, layout and other information. Then, LAM-GNN update the representation of nodes, so that nodes can fuse the features of their neighbors to better understand the context. Finally, we use the graph output by the last layer of GNN as the input of node and edge classifier. We apply an MLP as a node classifier, which maps each node's representation to the number of labeled classes. We use two fully connected layers as edge classifiers, and the edges have two types of labels: [pair] or [none].

3.1. XY Probability Graph Construction

The structure of the graph is crucial for subsequent graph learning. For instance, in a dense graph generated by full connection, GNNs focus on information from all nodes during the message passing. Undoubtedly, this places substantial stress on the model and contradicts the prior knowledge of document layout.

Our model is coarse-grained. Following the previous works of SelfDoc [27] and UniDoc [28], we use the region-level bounding boxes in our method. This can avoid an overly fine-grained with excessive contextualization and reduce redundant information.

We observe that both GraphDoc [25] and Doc-GCN [15] also emphasize locality in document understanding, meaning that blocks of text in a document are more dependent on their surrounding context. Inspired by the above methods, we design XYPG, which not only conforms to locality, but also solves the complex key-value relationships in the document. The algorithm flow chart of the method is shown in Algorithm 1.

Algorithm 1 XYPG Algorithm.

```
Input: Scanned document image $I$ with OCR results, contains $n$
    bounding boxes. Record the bounding box of the $i$-th area as
    $b_i = (x_{i0}, y_{i0}, x_{i2}, y_{i2})$.
Output: Undirected graph $G = (V, E, A)$.
 1: Let $v_i \leftarrow b_i, i = 1 \dots n$. So $V = (v_1, v_2 ... v_n)$;
 2: Initialize an list $E$ to store the generated edges.
 3: for $i = 1 \dots n$ do
 4:     Initialize an list $XY_{(i)}$ to store the XY nodes of $v_i$
 5:     for $j \in V, j \neq i$ do
 6:         Calculate $Is\_Orth(v_i, v_j)$
 7:         if $Is\_Orth(v_i, v_j)$ then
 8:             Add $v_j$ to $XY_{(i)}$
 9:         end if
10:     end for
11:     for $v_j \in XY_{(i)}$ do
12:         Calculate probabilities $P[v_i, v_j]$
13:         if $P[v_i, v_j] > threshold$ then
14:             Add $e_{ij} = (v_i, v_j)$ to $E$
15:         end if
16:     end for
17: end for
```

The main idea is to construct the edge set E based on the XY relationship and probability threshold. In the algorithm, $Is\_Orth(v_i, v_j)$ function determines whether nodes v_i and v_j are similar in the horizontal or vertical direction, that is, whether there is an XY relationship. List $XY_{(i)}$ is used to store the XY relationship nodes of node v_i. In lines 6-8 of the algorithm, we retrieve XY-related nodes for the node v_i. In line 12, we calculate the edge probability between the nodes v_i and v_j using formula (1). Subsequently, in lines 13-14, edges are created and added to the edge set E, where the $threshold$ is set at 70%.

$$P[v_i, v_j] = \frac{\sum_{j \in XY_{(i)}} Dist\left(v_i, v_j\right) - Dist\left(v_i, v_j\right)}{\sum_{j \in XY_{(i)}} Dist\left(v_i, v_j\right)}, \tag{1}$$

where $Dist(v_i, v_j)$ stands for computing the Manhattan distance of two nodes.

Node Feature Addition. After getting the XY probability graph, we fuse multi-modal features on nodes to comprehensively capture different types of information within the document. Record a document image as I, we get n bounding boxes after the OCR result, record the bounding box of the i-th area as b_i, and the contained text as t_i.

We use the pre-trained English model en_core_web_lg from the spaCy to obtain word embedding representation of text segment:

$$s_i = spaCy\left(t_i\right), 1 \leq i \leq n. \tag{2}$$

We choose to use the U-net [29] as our visual backbone and use the RoIAlign [30] to extract features relative to the bounding box of each entity:

$$v_i = ROIAlign\left(Unet\left(I\right), b_i\right), 1 \leq i \leq n. \tag{3}$$

In the geometric feature, note that the coordinates of the upper left corner of b_i are x_{i0}, y_{i0}, and the coordinates of the lower right corner are x_{i2}, y_{i2}, w_i and h_i are the size of image I:

$$l_i = \left[\frac{x_{i0}}{w_i}, \frac{y_{i0}}{h_i}, \frac{x_{i2}}{w_i}, \frac{y_{i2}}{h_i}\right], 1 \leq i \leq n. \tag{4}$$

The histogram feature h_i is generated through the t_i, which represents the percentage of literals, numbers, and other symbols in the text content.

We concat the four features to get the final node feature representation:

$$N\left(i\right) = Concat(P_1(s_i), P_2(v_i), P_3(l_i), P_4(h_i)), \tag{5}$$

where P_i, $i = 1, 2, 3, 4$ contains a linear layer, a norm layer and an activation function $Relu$. The purpose is to project different features to the same dimension to prevent information imbalance.

In VrDU, t_i captures the semantic and grammatical information of the text, g_i is used to understand the typesetting style of the document, v_i can capture the visual information in the document such as color, shape, etc., and h_i complements the similarity of the text.

3.2. Layout Attention Mechanism

Message passing is a general framework and programming paradigm for implementing GNNs. It summarizes various GNNs models from the perspective of aggregation and updating. Let x_i be the feature for node v_i, and w_e be the feature for edge $e = (v_i, v_j)$. The original message passing defines the following edge-wise and node-wise computation:

$$m_e^{(t+1)} = \phi\left(x_i^{(t)}, x_j^{(t)}, w_e^{(t)}\right), \quad e = (v_i, v_j) \in E, \tag{6}$$

$$x_i^{(t+1)} = \psi\left(x_i^{(t)}, \rho_{\ j \in N_{(i)}} \left\{m_e^{(t+1)} : e = (v_i, v_j) \in E\right\}\right), \tag{7}$$

where ϕ is a message function defined on each edge to generate a message by combining the edge feature with the features of its incident nodes. ψ is an update function defined on each node to update the node feature by aggregating its incoming messages using the reduce function ρ.

Popular GNNs frameworks such as GCN, GraphSAGE, GAT and GIN [31] all follow this paradigm. Some of them regard all neighbor nodes as having the same weight during message passing. However, in document understanding, different text fragments may have different contributions in meaning and context, so all neighbor nodes are simply regarded as equivalent is inappropriate.

To address the above issues, in this section we propose the layout attention mechanism. We introduce an attention mechanism to dynamically assign the weights of neighbor nodes, which can make neighbor nodes with higher similarity or closer association have greater influence in message passing. In this way, GNNs can better capture the semantic relationship between nodes, thereby improving the performance in tasks such as document understanding.

We redefine the message passing as:

$$x_i^{(t+1)} = Concat(x_i^{(t)}, \textstyle\sum_{j \in N_{(i)}} x_j^{(t)} \cdot w_e[v_i, v_j]), \tag{8}$$

where $w_e[v_i, v_j]$ is the attention weight of nodes v_i and v_j calculated by the layout attention mechanism. For each node $v_i \in V$, the neighbor set of node v_i is obtained: $N_{(i)} = v_j | (v_i, v_j) \in E$, and the attention weight are calculated as follow:

$$w_e\left[v_i, v_j\right] = \frac{\sigma\left(polar\left(v_i, v_j\right)\right)}{\sum_{j \in N_{(i)}} \sigma\left(polar\left(v_i, v_j\right)\right)}, \tag{9}$$

where $polar(v_i, v_j)$ calculates the polar coordinate distance between nodes v_i and v_j. σ is the activation function, ensure that it is a monotonically decreasing function to ensure close distance nodes can be assigned larger weights.

We choose to use polar coordinate distance to calculate the distance between bounding boxes. Because it takes into account angle and directional features, which are closely related to the layout of the document and better align with the requirements of KIE and DLA. We can capture the relative positions and directional relationships between different elements in the document, thereby improving the accuracy of downstream tasks.

4. EXPERIMENTS

4.1. Datasets

In this section, we introduce three datasets used to evaluate the SEG-Doc model: the FUNSD [21] dataset and SROIE [22] dataset for the KIE task, and the RVL-CDIP [23] dataset for the DLA task. We conducted a series of experiments on these datasets to comprehensively verify the effectiveness and performance of our proposed model.

FUNSD. FUNSD is a dataset for form understanding. The dataset contains 199 real, fully annotated, scanned forms. Documents are noisy, and forms vary greatly in appearance. The dataset is divided into 149 training data and 50 testing data. We focus on the semantic entity labeling (SEL) and entity labeling (EL) tasks in this paper.

SROIE. The SROIE dataset (Task 3) aims to extract texts of several key fields from receipts. It comprises 626 receipts for training and 347 receipts for testing. Each receipt contains four crucial text fields: company, address, date, and total. We use the official OCR annotations in our experiments.

RVL-CDIP. In the work of Riba et al. [23], they released a dataset which is another subset of RVL-CDIP. The authors specifically selected 518 documents from the invoices classes. These samples are annotated for six different regions: invoice_info, positions, receiver, supplier, total and other. We focus on layout analysis and table detection tasks in the dataset.

4.2. Implementation Details

We use K-Fold cross-validator to partition the training set on all datasets. The parameter random_seed is uniformly set to 42. The parameter n_splits is 10 on FUNSD, 7 on SROIE and RVL-CDIP. We report the average results of the experiments.

We utilize the pre-trained English model en_core_web_lg 3.3.0 from the spaCy to extract text features (dim = 300). A pre-trained U-net on FUNSD is employed to obtain visual features for each bounding box (dim = 1448). Geometric and histogram features (dim = 4) are presented in Section 3.1. During feature fusion, we project each modal feature to dim = 300 using fully connected layer. In the ablation study in Section 4.4, according to the experimental results in Table 6, we choose GraphSAGE using the Mean aggregator as our graph learning network. The hidden size is set to 300, and the number of recurrent layers is 3.

The proposed model is trained on 1 NVIDIA 3090 GPU with 24 GB memory. Our model is trained from scratch using Adam as the optimizer and the batch size is 15 at the training phase. The learning rate is set to 1×10^{-3} and the weight decay is set to 1×10^{-4} over the whole training phase. We choose cross-entropy loss as the loss function. We use EarlyStopping to monitor the model during training to prevent model overfitting and the patience is set to 2000 epochs.

4.3. Comparisons With The SOTAs

In this section, we compare our method with other state-of-the-art methods on three datasets. We choose GraphSAGE using the Mean aggregator as our backbone. Preventing the influence of randomness or chance, we report the average of K-Fold experiments as the final results.

In form understanding, Table 1 displays the F1 scores for both semantic entity labeling (SEL) and entity linking (EL) tasks on the FUNSD dataset. It can be seen that on the SEL task, LayoutLMv3$_{\text{BASE}}$ achieved the best result with 90.29%, and DocTr ranked second with 84.00%. Our method ranked third with 83.22%, which is only 0.78% lower than DocTr. In particular, our method outperforms all the baseline models on EL task.

Table 1. Comparison with state-of-the-art methods on the FUNSD dataset. **Bold** indicates the SOTA, underlined indicates the second best.

Method	GNN	F1(↑)		# Params
		SEL	EL	
BERT$_{\text{BASE}}$ [32]	×	60.26	27.65	110M
BROS$_{\text{BASE}}$ [8]	×	83.05	71.46	139M
LayoutLM$_{\text{BASE}}$ [5]	×	78.66	45.86	113M
LayoutLMv2$_{\text{BASE}}$ [6]	×	82.76	42.91	200M
LayoutLMv3$_{\text{BASE}}$ [10]	×	**90.29**	-	133M
DocTr [33]	×	84.00	73.90	153M
FUDGE [24]	✓	66.52	56.62	-
Doc2Graph [17]	✓	82.25	53.36	6.2M
SEG-Doc(ours)	✓	83.22	**82.50**	18.9M

In receipt understanding, Table 2 displays the experimental results on the SROIE dataset. We achieved the best result with 97.98%. We can beat models with more parameters than we have, which is our greatest advantage. The results show that our model can compete with state-of-the-art methods in the field.

Table 2. Comparison with state-of-the-art methods on the SROIE dataset. **Bold** indicates the SOTA, underlined indicates the second best.

Method	GNN	F1(↑)	# Params
BERT$_{\text{BASE}}$ [32]	×	90.99	110M
BROS$_{\text{BASE}}$ [8]	×	95.48	139M
LayoutLM$_{\text{BASE}}$ [5]	×	94.38	113M
LayoutLMv2$_{\text{BASE}}$ [6]	×	96.25	200M
GraphIE [11]	✓	94.46	-
PICK [14]	✓	96.12	-
SEG-Doc(ours)	✓	**97.98**	18.9M

In invoice understanding, Table 3 and Table 4 show the experimental results of tasks layout analysis and table detection on the RVL-CDIP dataset respectively. It can be seen that our method achieves the new SOTA on both tasks. Especially on the table detection task, we outperform the previous best work by about 6 percentage points.

In the SEG-Doc model, We effectively fuse multi-modal features. The XY probability graphs constructed by the XYPG module better align with the structural aspects of real-world documents. The LAM enhances the influence of neighboring nodes with stronger correlations during the message passing process. Therefore, we achieve

Table 3. Accurary comparison for layout analysis task on the RVL-CDIP dataset. **Bold** indicates the SOTA, underlined indicates the second best.

Method	Accuracy(↑)	
	Max	Mean
Riba et al. [23]	62.30	-
Doc2Graph [17]	69.80	67.80
SEG-Doc(ours)	**70.42**	**69.70**

Table 4. Accurary comparison for table detection task on the RVL-CDIP dataset. **Bold** indicates the SOTA, underlined indicates the second best.

Method	threshold	Metrics(↑)		
		precision	recall	F1
Riba et al. [23]	0.1	0.2520	0.3960	0.3080
Riba et al. [23]	0.5	0.1520	0.3650	0.2150
Doc2Graph [17]	0.5	0.3786	0.3723	0.3754
SEG-Doc(ours)	0.5	**0.4411**	**0.4355**	**0.4383**

excellent experimental results comparable to SOTAs. In particular, we can beat models with more parameters than we have, which is our greatest advantage.

4.4. Ablation Study

Our ablation study is experimented on the FUNSD dataset.

Effectiveness of XYPG. Table 5 shows the effectiveness of our XYPG module. Compared with the full connection and KNN method, the full connection graphs usually introduce redundant connection and parameters, the KNN graph only considers the distance and ignores the structural characteristics of the document. While the graph constructed by XYPG can better express the essence of the document, and we have solved the problems caused by the complex table structure in the document. In Table 5, we choose GraphSAGE using the Mean aggregator as our backbone.

Table 5. Accuracy comparison between XYPG and different graph construction methods on the FUNSD dataset. **Bold** indicates the SOTA, underlined indicates the second best.

Method	F1(↑)	
	SEL	EL
KNN=2	81.21	56.84
KNN=5	81.50	47.64
KNN=10	80.50	40.96
Full Connection	81.65	47.77
XYPG(ours)	**83.22**	**82.50**

Effectiveness of LAM. Table 6 shows the effectiveness of the layout attention mechanism. We choose GCN, GraphSAGE and GIN for experiments. Note that on GraphSAGE we have selected three different types of aggregators: Mean, GCN, LSTM. It can be seen that LAM has improved the results of the above networks on FUNSD. By dynamically assigning neighbor weights, LAM can make neighbor nodes with higher similarity have greater influence in message passing. Therefore, our method can learn more powerful node representations, which is crucial for downstream tasks. Based on the parameter amount and experimental results, we choose GraphSAGE using the Mean aggregator as our final solution.

Table 6. Accuracy comparison of different graph neural networks after integrating LAM on the FUNSD dataset. **Bold** indicates the SOTA, underlined indicates the second best.

Network	LAM	F1(↑)		# Params
		SEL	EL	
GCN	-	79.65	74.88	10.3M
	✓	80.07	77.56	
GraphSAGE(Mean)	-	81.97	80.36	18.9M
	✓	**83.22**	**82.50**	
GraphSAGE(GCN)	-	79.63	77.33	10.3M
	✓	81.41	78.80	
GraphSAGE(LSTM)	-	82.24	79.89	157.2M
	✓	82.63	82.11	
GIN	-	79.12	74.27	10.3M
	✓	80.81	76.97	

5. CONCLUSION

In this paper, we propose SEG-Doc framework for the KIE and DLA tasks, which involves the XY probability graph (XYPG) method and the layout attention mechanism (LAM). Specifically, XYPG can generate graph structures that reflect real-world documents, LAM can be integrated into GNNs to learn richer node representations. SEG-Doc fully utilizes the multi-modal features in a document. Experimental results show that we have achieved excellent results comparable to strong baselines on the FUNSD, SROIE and RVL-CDIP datasets, and even significantly better than strong baselines on some tasks. In particular, our model has small parameters and fast training, which makes it more practical in industry.

6. REFERENCES

[1] Zhenrong Zhang, Jianshu Zhang, Jun Du, and Fengren Wang, “Split, embed and merge: An accurate table structure recognizer,” *Pattern Recognition*, vol. 126, pp. 108565, 2022.

[2] Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei, “Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding,” *arXiv preprint arXiv:2104.08836*, 2021.

[3] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal, “Unifying vision, text, and layout for universal document processing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19254–19264.

[4] Pengcheng Guo, Yonghong Song, Yongbiao Deng, Kangkang Xie, Mingjie Xu, Jiahao Liu, and Haijun Ren, “Dcmai: A dynamical cross-modal alignment interaction framework for document key information extraction,” *IEEE TCSVT*, 2023.

[5] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou, “Layoutlm: Pre-training of text and layout for document image understanding,” in *KDD*, 2020.

[6] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al., "Layoutlmv2: Multi-modal pre-training for visually-rich document understanding," *IJCNLP*, 2021.

[7] Peng Zhang, Can Li, Liang Qiao, Zhanzhan Cheng, Shiliang Pu, Yi Niu, and Fei Wu, "Vsr: a unified framework for document layout analysis combining vision, semantics and relations," in *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I 16*. Springer, 2021, pp. 115–130.

[8] Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park, "Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents," in *AAAI*, 2022.

[9] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha, "Docformer: End-to-end transformer for document understanding," in *ICCV*, 2021.

[10] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei, "Layoutlmv3: Pre-training for document ai with unified text and image masking," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 4083–4091.

[11] Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay, "Graphie: A graph-based framework for information extraction," *arXiv preprint arXiv:1810.13083*, 2018.

[12] Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao, "Graph convolution for multimodal information extraction from visually rich documents," *arXiv preprint arXiv:1903.11279*, 2019.

[13] Manuel Carbonell, Pau Riba, Mauricio Villegas, Alicia Fornés, and Josep Lladós, "Named entity recognition and relation extraction with graph neural networks in semi structured documents," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9622–9627.

[14] Wenwen Yu, Ning Lu, Xianbiao Qi, Ping Gong, and Rong Xiao, "Pick: processing key information extraction from documents using improved graph learning-convolutional networks," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 4363–4370.

[15] Siwen Luo, Yihao Ding, Siqu Long, Josiah Poon, and Soyeon Caren Han, "Doc-gcn: Heterogeneous graph convolutional networks for document layout analysis," *arXiv preprint arXiv:2208.10970*, 2022.

[16] Guozhi Tang, Lele Xie, Lianwen Jin, Jiapeng Wang, Jingdong Chen, Zhen Xu, Qianying Wang, Yaqiang Wu, and Hui Li, "Matchvie: Exploiting match relevancy between entities for visual information extraction," *arXiv preprint arXiv:2106.12940*, 2021.

[17] Andrea Gemelli, Sanket Biswas, Enrico Civitelli, Josep Lladós, and Simone Marinai, "Doc2graph: a task agnostic document understanding framework based on graph neural networks," in *ECCV*. Springer, 2022, pp. 329–344.

[18] Rinon Gal, Shai Ardazi, and Roy Shilkrot, "Cardinal graph convolution framework for document information extraction," in *Proceedings of the ACM Symposium on Document Engineering 2020*, 2020, pp. 1–11.

[19] Will Hamilton, Zhitao Ying, and Jure Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[20] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2016.

[21] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran, "Funsd: A dataset for form understanding in noisy scanned documents," in *ICDAR Workshop*, 2019.

[22] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar, "Icdar2019 competition on scanned receipt ocr and information extraction," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1516–1520.

[23] Pau Riba, Anjan Dutta, Lutz Goldmann, Alicia Fornés, Oriol Ramos, and Josep Lladós, "Table detection in invoice documents by graph neural networks," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 122–127.

[24] Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, and Curtis Wiginton, "Visual fudge: Form understanding via dynamic graph editing," in *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I 16*. Springer, 2021, pp. 416–431.

[25] Zhenrong Zhang, Jiefeng Ma, Jun Du, Licheng Wang, and Jianshu Zhang, "Multimodal pre-training based on graph attention network for document understanding," *IEEE TMM*, 2022.

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

[27] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu, "Selfdoc: Self-supervised document representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5652–5660.

[28] Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmpalios, Ani Nenkova, and Tong Sun, "Unidoc: Unified pretraining framework for document understanding," *Advances in Neural Information Processing Systems*, vol. 34, pp. 39–50, 2021.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[30] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[31] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018.

[32] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[33] Haofu Liao, Aruni RoyChowdhury, Weijian Li, Ankan Bansal, Yuting Zhang, Zhuowen Tu, Ravi Kumar Satzoda, R Manmatha, and Vijay Mahadevan, "Doctr: Document transformer for structured information extraction in documents," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19584–19594.