Neural Collaborative Graph Machines for Table Structure Recognition - Supplementary Materials

Hao Liu[†*] Xin Li[*] Bing Liu Deqiang Jiang Yinsong Liu Bo Ren
Tencent YouTu Lab
{ivanhliu, fujikoli, billbliu, dqiangjiang, jasonysliu, timren}@tencent.com

A. Feature Extraction

A.1. Multi-modality Features

Geometry embedding. We derive the geometry feature of each text segment bounding box as $\left(\frac{x}{W}, \frac{y}{H}, \frac{w}{W}, \frac{h}{H}\right)^\top$, where W and H are the width and height of the table image. (x, y) represents the center point of the box while height h and width w correspond to its short side and long side respectively. Then a d-dimension Fully-Connected (FC) layer is applied on the above vectors to obtain the geometry embeddings $\mathbf{F}^{\mathrm{G}} = \{\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_N\} \in \mathbb{R}^{N \times d}$.

Appearance embedding. We employ ResNet18-based CNN [7] as backbone to extract whole table image feature. In detail, the backbone consists of *conv1* to *conv2_2* of ResNet18 followed by three convolutional layers of size $3 \times 3 \times 64$. Hereafter, the output of backbone is applied by the RoI Align [6] in terms of text segment bounding boxes. After passing a FC layer with d dimensions, appearance embeddings $\mathbf{F}^{\mathrm{A}} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N\} \in \mathbb{R}^{N \times d}$ are obtained.

Content embedding. First, we embed corresponding text of each text segment bounding box in distributional space via word2vec [2]. Then, one convolutional layer with $7 \times 1 \times d$ kernel size and 1 stride is applied to model text sequential feature as content feature embeddings $\mathbf{F}^{\mathrm{C}} = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\} \in \mathbb{R}^{N \times d}$.

A.2. Ablation Study of Mutil-modalities

As shown in Tab. 1, we observe that among the three modalities, "G" plays a dominant role, followed by "A", and finally "C". The proposed model leveraging all three modalities can achieve impressive progress under all evaluation metrics. In addition, we also explore the attention weights of individual modality. That is, the attention weights of "A" and "G" tend to be grid-like, indicating that the model focuses on the spatial position of the row or column in global range. And the attention weights of "C" are inclined to emphasize on local successive segment bounding boxes. To sum up, the inductive biases of different modalities are of large disparency.

Input Modality			Setup-B		
A	G	C	P	R	F1
✓	✗	✗	89.8	47.9	62.5
✗	✓	✗	97.9	97.7	97.8
✗	✗	✓	70.5	39.0	50.2
✓	✓	✗	98.6	98.3	98.4
✗	✓	✓	98.0	95.0	96.5
✓	✗	✓	87.6	89.3	88.4
✓	✓	✓	**98.8**	**99.3**	**99.0**

Table 1. Ablation studies of multi-modalities on SciTSR-COMP dataset. "A", "G" and "C" stand for "appearance", "geometry" and "content" modality respectively.

B. Multi-head Attention

We build the core collaborative block of our method upon Multi-head Attention (MHA) [17] module. Here, we briefly introduce it as preliminary knowledge. Given queries $\mathbf{Q}$, keys $\mathbf{K}$ and values $\mathbf{V}$, MHA is defined as:

$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_h)\mathbf{W}^*,$$
$$\mathbf{H}_i = Attention(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), i \in \{1, 2, ..., h\},$$
$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}})\mathbf{V},$$

where d_k is the dimension of keys while h is the head number. $\mathbf{W}_i^Q \in \mathbb{R}^{d_m \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_m \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_m \times d_v}$ and $\mathbf{W}_i^* \in \mathbb{R}^{hd_v \times d_m}$ are projection matrices separately. Essentially, the attention process can be regarded as "memory accessing" procedure.

[*]Equal contribution. [†]Contact person.

C. Training Strategy

C.1. Design of Loss Function

The binary classification loss is widely applied in previous graph-based works of table structure recognition (TSR). Particularly, we train our proposed Neural Collaborative Graph Machines (NCGM) in an end-to-end way to satisfy both the contrastive objective and to predict belonging classes of the output embedding pairs. Given a pair of collaborative graph embeddings ($\{\mathbf{e}_{(a)}, \mathbf{e}_{(b)}\}$) and corresponding concatenated vector $\mathbf{u}_{(a,b)}$, we define the multi-task loss function as:

$$
\begin{aligned}
&\mathcal{L} = \mathcal{L}_{cell} + \mathcal{L}_{col} + \mathcal{L}_{row}, \\
&\mathcal{L}_{\sim} = \lambda_1 \mathcal{L}_{class} + \lambda_2 \mathcal{L}_{con}, \\
&\mathcal{L}_{con} = \left\| \mathbf{e}_{(a)} - \mathbf{e}^{+}_{(b)} \right\|_2^2 + \max\left\{0, \alpha - \left\| \mathbf{e}_{(a)} - \mathbf{e}^{-}_{(b)} \right\|_2^2 \right\}, \\
&\mathcal{L}_{class} = -\log(P(z = c|\mathbf{u}_{(a,b)})), \\
&P(z = c|\mathbf{u}_{(a,b)}) = \frac{\exp(S_c \mathbf{u}_{(a,b)})}{\sum_k \exp(S_k \mathbf{u}_{(a,b)})}, c \in \{0, 1\},
\end{aligned}
$$

where $\mathcal{L}_{\sim}$ represents $\mathcal{L}_{cell}$, $\mathcal{L}_{col}$ or $\mathcal{L}_{row}$, corresponding to cell, column and row relationship loss. $\mathcal{L}_{con}$ is contrastive loss in which $\mathbf{e}^{+}_{(b)}$ and $\mathbf{e}^{-}_{(b)}$ are the positive and negative pair of $\mathbf{e}_{(a)}$ respectively. The margin parameter α is set to 1. Correspondingly, $\mathcal{L}_{class}$ is the standard softmax loss in terms of $\mathbf{u}_{(a,b)}$. z is the predicted class for the input pairs, and S is the weight matrix used in the softmax function, and S_c and S_k represent the c-th and k-th column of it, respectively. $c = 1$ denotes the concatenated pairs belong to the same cell/column/row, and otherwise $c = 0$. They are combined by weight parameters λ_1 and λ_2. Considering memory efficiency, we also introduce Monte Carlo sampling for constructing collaborative graph embedding pairs in the training phase, which is similar to [12]. For inference, the sampling is not performed and we construct all collaborative graph embeddings as pairs.

C.2. Forward Process

For clarity, the detailed forward process of NCGM is shown in Alg. 1. Note, the symbol with superscript "$\sim$" denotes it is derived from "appearance", "geometry" or "content" modality. And the symbol with subscript "$\sim$" represents it belongs to one of "cell", "column" or "row" relationships. The sample size S of Monte Carlo sampling is set to 10 in the training phase.

C.3. Ablation Study of Loss

We also perform experiments to evaluate the effect of different loss functions. For the sake of fairness, all models with different loss settings are trained with the same backbone model and training data. As shown in Tab. 2,

Algorithm 1 NCGM pseudo code.

Input: $\mathbf{T}$, $\mathbf{GT}_{\sim}$; // $\mathbf{T}$ denotes input table elements. $\mathbf{GT}_{\sim}$ ($\mathbf{GT}_{\sim} \in \{\mathbf{GT}_{cell}, \mathbf{GT}_{row}, \mathbf{GT}_{col}\}$) represents the Ground Truth of different relationships.
Output: $\mathbf{F}^{pred}_{\sim}$

/* Extract features by Compressed Multi-head Attention. */
Function `CMHA` ($\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$)**:**
  $\mathbf{Y} \leftarrow MHA(\mathbf{Q}, MC(\mathbf{K}), MC(\mathbf{V}))$
  return $\mathbf{Y}$

/* Ego Context Extractor. */
Function `ECE` ($\mathbf{C}^{\sim}_{(l-1)}$)**:**
  $\mathbf{Q} \leftarrow \mathbf{C}^{\sim}_{(l-1)}$
  $\mathbf{K} \leftarrow \mathbf{V} \leftarrow \mathbf{H}^{\sim}_{\Theta} \leftarrow h_{\Theta}(\mathbf{x}_i, \mathbf{x}_j)$
  $\mathbf{C}^{\sim}_{(l)} \leftarrow CMHA(\mathbf{Q}, \mathbf{K}, \mathbf{V})$
  return $\mathbf{C}^{\sim}_{(l)}$

/* Cross Context Synthesizer. */
Function `CCS` ($\mathbf{M}^{C}_{(l-1)}$, $\mathbf{C}^{A}_{(l)}$, $\mathbf{C}^{G}_{(l)}$)**:**
  $\mathbf{Q} \leftarrow \mathbf{M}^{C}_{(l-1)}$
  $\mathbf{K} \leftarrow \mathbf{V} \leftarrow \mathbf{C}^{A}_{(l)} \textcircled{\cup} \mathbf{C}^{G}_{(l)}$
  $\mathbf{M}^{C}_{(l)} \leftarrow CMHA(\mathbf{Q}, \mathbf{K}, \mathbf{V})$
  return $\mathbf{M}^{C}_{(l)}$

Function `Main`**:**
  $\mathbf{F}^{\sim} \leftarrow$ Extract appearance, geometry and content features from $\mathbf{T}$.
  /* Initialization. */
  $\mathbf{C}^{\sim}_{(0)} \leftarrow \mathbf{M}^{\sim}_{(0)} \leftarrow \mathbf{F}^{\sim}$
  /* Generate collaborative embeddings by NCGM. */
  for *l = 1, 2, 3* **do**
    $\mathbf{C}^{A}_{(l)} \leftarrow ECE(\mathbf{C}^{A}_{(l-1)})$
    $\mathbf{C}^{G}_{(l)} \leftarrow ECE(\mathbf{C}^{G}_{(l-1)})$
    $\mathbf{C}^{C}_{(l)} \leftarrow ECE(\mathbf{C}^{C}_{(l-1)})$
    $\mathbf{M}^{A}_{(l)} \leftarrow CCS(\mathbf{M}^{A}_{(l-1)}, \mathbf{C}^{G}_{(l)}, \mathbf{C}^{C}_{(l)})$
    $\mathbf{M}^{G}_{(l)} \leftarrow CCS(\mathbf{M}^{G}_{(l-1)}, \mathbf{C}^{A}_{(l)}, \mathbf{C}^{C}_{(l)})$
    $\mathbf{M}^{C}_{(l)} \leftarrow CCS(\mathbf{M}^{C}_{(l-1)}, \mathbf{C}^{A}_{(l)}, \mathbf{C}^{G}_{(l)})$
  $\mathbf{E} \leftarrow \mathbf{M}^{A}_{(3)} \oplus \mathbf{M}^{G}_{(3)} \oplus \mathbf{M}^{C}_{(3)}$
  /* Construct pairs. */
  $\mathbf{U} \leftarrow Pairing(\mathbf{E})$
  if *train* **then**
    /* Monte Carlo sampling. S is the sample size. */
    $\left[\mathbf{U}^{S}; \mathbf{GT}^{S}_{\sim}\right] \leftarrow Sampling([\mathbf{U}; \mathbf{GT}_{\sim}], S)$
    /* Separately compute cell/col/row loss. */
    $\mathcal{L}_{\sim} \leftarrow Loss(\mathbf{U}^{S}, \mathbf{GT}^{S}_{\sim})$
    Backward.
  else
    /* Separately predict cell/col/row relationships. */
    $\mathbf{F}^{pred}_{\sim} \leftarrow Classify_{\sim}(\mathbf{U})$
  return

Loss Function		Setup-B		
$\mathcal{L}_{class}$	$\mathcal{L}_{con}$	P	R	F1
✓	✗	**98.9**	98.6	98.7
✗	✓	94.4	92.1	93.2
✓	✓	98.8	**99.3**	**99.0**

Table 2. Ablation studies of losses on SciTSR-COMP dataset. $\mathcal{L}_{con}$ and $\mathcal{L}_{class}$ are contrastive loss and binary classification loss respectively.

we observe that the model trained by binary classification loss $\mathcal{L}_{class}$ outperforms the one trained by contrastive loss $\mathcal{L}_{con}$, while the combination of $\mathcal{L}_{class}$ and $\mathcal{L}_{con}$ can achieve better performance than either of the two. We attribute this to the extra regularization provided by contrastive loss, that makes the model pay more attention to hard negative pairs. As a consequence, our method can learn more discriminative representations of row, column or cell relationships.

D. Post-processing

For a fair comparison with other methods, we perform post-processing on the results of our method. As opposed to pre-processing, post-processing aims to convert the adjacency matrix containing relationships to spanning information either in "XML" format for evaluating physical structure recognition or "HTML" format for evaluating logical structure recognition respectively, which is shown in Fig. 1.

Post-process for physical structure recognition. We also take the row relationship for example. First of all, all boxes are sorted by their y coordinates of top left points to generate their indexes (represented in blue). For each box v_i, the row belonging list is generated according to row adjacency matrix. Afterwards, the spanning information in "XML" format can be obtained. Here, we define the table box row index according to the boundaries of boxes, as illustrated by the red numbers in Fig. 1. In detail, boxes belonging to the same row belonging list are assigned with the same starting-row and ending-row indexes. Similarly, we can also obtain the spanning results from column adjacency matrix. Finally, an XML file is created with the extracted spanning information along with bounding box coordinates and contents.

Post-process for logical structure recognition. As for the datasets (*i.e.*, TableBank [9] and PubTabNet [18]) in which GTs are in the form of HTML sequences, the evaluation protocol put more emphasis on correctly recognizing the logical structure of tables. We can also convert the adjacency matrix of relationship to HTML tag sequences according to the belonging list.

E. Datasets

E.1. Datasets for Experiments

We perform large-scale experiments on various benchmark datasets as summarized in Tab. 3. Among, ICDAR-2013 [5], ICDAR-2019 [4], UNLV [15], WTW [11], SciTSR [1] and SciTSR-COMP [1] are employed for physical structure recognition, while TableBank [9] and PubTabNet [18] are adopted for evaluating logical structure recognition performance.

In particular, it should be noted that there exists no training set in ICDAR-2013 [5] and UNLV [15] datasets, so we extend the two datasets to the partial versions (*i.e.*, ICDAR-2013-P and UNLV-P). Concretely, we randomly split each dataset into five folds, of which four folds for training and the left one for testing. The random splits are performed ten rounds for computing averaged performance, which is similar to TabStruct-Net [14].

For more clarity, we also count the number of text segment bounding boxes and tables in every table image for different datasets in Tab. 3 ("-" means no training set provided).

Dataset	Train		Test		Image	Content	C-Box	T-Box
	Table (Amt)	Box (Avg)	Table (Amt)	Box (Avg)				
IC13	-	-	158	93	✓	✓	✗	✓
IC13-P	124	92	34	96	✓	✓	✗	✓
IC19	600	314	150	359	✓	✗	✓	✗
UNLV	-	-	558	77	✓	✗	✓	✗
UNLV-P	446	84	112	43	✓	✗	✓	✗
WTW	10970	101	3611	96	✓	✗	✓	✗
Sci.	12000	47	3000	48	✓	✓	✗	✓
Sci.-C	12000	47	716	74	✓	✓	✗	✓
Sci.-C-A	24000	47	1432	74	✓	✓	✗	✓
TableBank	145K	50	1000	49	✓	✗	✗	✗
PubTabNet	339K	72	114K	74	✓	✓	✗	✓

Table 3. Statistics of the datasets our experiments performed on. "Amt" and "Avg" denote "Amount" and "Average" separately. "-P" means partial dataset and "-A" represents augmented dataset by distortion. "IC13", "IC19", "Sci." and "Sci.-C" are short for "ICDAR-2013", "ICDAR-2019", "SciTSR" and "SciTSR-COMP" individually. "C-Box" and "T-Box" stand for "cell bounding boxes" and "text segment bounding boxes" respectively.

E.2. Processing on Inconsistent Annotation Levels

Pre-process for bounding boxes. One major challenge of performing comparisons on different datasets lies in the inconsistency of annotation levels on the bounding boxes. As shown in Tab. 3, ICDAR-2019 [4], UNLV [15] and WTW [11] datasets have ground truth (GT) bounding boxes of cell, while ICDAR-2013 [5] and SciTSR [1] datasets take text segment bounding boxes as GT annotations. In our method, we regard text segment bounding boxes as table

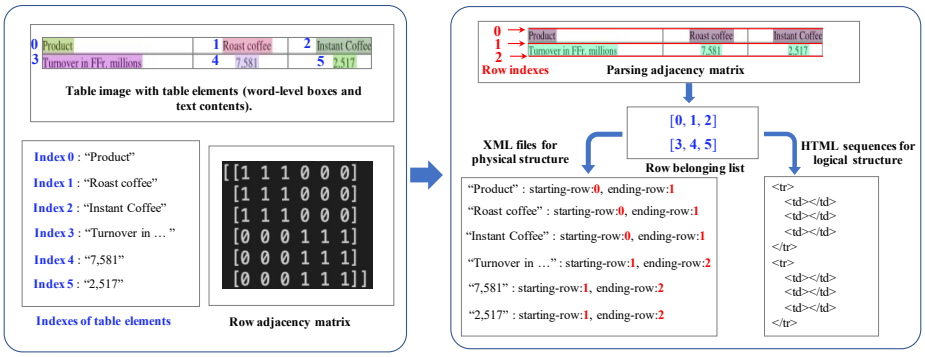


Figure 1. Post-processing of our proposed NCGM.

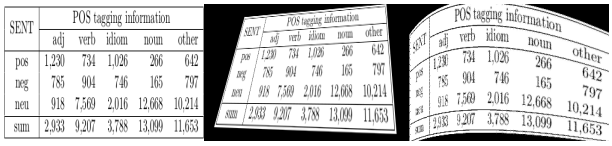
elements. Therefore, we do some processing to eliminate the inconsistency in annotation levels.

In detail, we convert the cell bounding boxes to the text segment ones according to OCR results in the training stage. For the text-segment-level datasets (*i.e.*, ICDAR-2013 [5] and SciTSR [1]), we consider the original boxes and text contents as model input directly, which are extracted by parsing GT files. To unify the input format, for the cell-level datasets (*i.e.*, ICDAR-2019 [4], UNLV [15] and WTW [11]), the text-segment-level boxes with contents are generated by the OCR results of Tesseract [16]. Note that an original cell-level box may contain more than one text-segment-level boxes, which have the common row and column spanning information (*i.e.*, starting-row, starting-column, ending-row and ending-column indexes) of the corresponding cell-level box. During the testing time, however, we still keep the original cell-level or text-segment-level boxes as GTs instead of the pre-processed ones in Setup-B, which ensures consistency while comparing our method against previously published ones. Especially, we take the result boxes of detection in FLAG-Net [10] and the OCR results of Tesseract [16] as inputs for fair comparison in Setup-A.

Pre-process for relationships. In order to provide the uniform GT of adjacency relationships ($\mathbf{GT}_{\sim}$ in Alg. 1) for the model's training phase, we convert the spanning information of table's rows and columns in various formats into the adjacency matrices of cell, row and column, which represent three adjacency relationships for the table elements. Take the row adjacency matrix for example, if the i-th and j-th boxes belong to the same row relationship, the value located at (i, j) in adjacency matrix is assigned to 1, otherwise to 0. In this way, we can construct the row adjacency matrix to represent the relationship of row. The adjacency matrices of cell and column are also generated in the similar way.

E.3. Synthesizing Method

To further investigate the capacity of TSR methods under more challenging scenes, we augment existing datasets with the following two kinds of image distortion algorithms to simulate distractors brought by capture device, which are visualized in Fig. 2.



(a) Original Image (b) Distortion 1 (c) Distortion 2

Figure 2. Images from SciTSR-COMP dataset applied by distortion algorithms.

Distortion 1. The first disortion is based on perspective transformation algorithm, which projects the table image to a new view plane according to the mapping matrix, as is shown in Fig. 2(b).

Distortion 2. For the second kind of distortion, we employ a algorithm based on the quadratic Bézier curve [8] to augment the datasets, which can be defined as:

$$B_2(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, t \in [0, 1],$$

where P_0, P_1 and P_2 denote three control points of the Bézier curve.

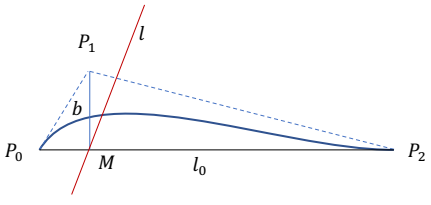


Figure 3. Determination of control points in Bézier curve.

Concretely, for each row of the image, we generate a quadratic Bézier curve applied on it to implement pixel-level distortion. There are three main steps to determine the control points of quadratic Bézier curve. As shown in Fig. 3, we first randomly initialize the axis line l (the red line) and the offset b. Next, each row of the image is regarded as l_0, and its starting point is deemed as the control point P_0 while ending point as P_2. Besides, the control point P_1 is located at a position offset from M (the intersection point between l_0 and l) by b. Through this way, the quadratic Bézier curves are determined by the control points, which are applied on each row of image pixels to perform distortion. It is worth mentioning that the blank pixels generated in the distortion process are interpolated by neighbouring pixels.

F. Computational Complexity

To further compare the computational complexity of existing various methods of table structure recognition, we summarize the model sizes and the inference operations of different models in Tab. 4. Since LGMPA [13] and CycleCenterNet [11] recover table structure based on heuristic rules after detecting cells, which is infeasible to perform the comparison between them and our method, we do not report them in Tab. 4. In particular, note that TabStruct-Net [14] and FLAG-Net [10] are only tested for structure recognition, so we do not count the parameters and operations of cell detection for a fair comparison.

Although the parameters and FLOPs of NCGM are larger than FLAG-Net [10], the performance of our method increases average F1-score by a large margin especially under challenging scenarios (*e.g.*, WTW and SciTSR-COMP-A). The reasons for increasing computational complexity is probably because of the individual operations on multiple modalities in our method. Compared with TabStruct-Net [14], NCGM can achieve better performance with less parameters and similar computational budgets. Moreover, the model size and FLOPs of GraphTSR [1] are the smallest among the compared methods, but it only utilizes the box coordinates as input to recognize table structure, which cannot achieve comparable performance than other methods. We consider to optimize the computational complexity and size of model without performance degradation in the future work.

G. Jensen-Shannon Divergence

We in this work introduce the Jensen-Shannon Divergence [3] to measure the average diversity of attention maps in CCS, which is defined as:

$$JSD = H(\frac{1}{n}\sum_{i=1}^{n}\mathbf{P}_i) - \frac{1}{n}\sum_{i=1}^{n}H(\mathbf{P}_i),$$

Method	Setup-B	
	#Param	FLOPs
GraphTSR [1]	7.0e-4	1.8e-4
DGCNN [12]	0.8	4.1
TabStruct-Net [14]	4.7	11.9
FLAG-Net [10]	1.9	3.3
NCGM	3.1	12.7

Table 4. Computational complexity comparison of different methods. #Param denotes the number of parameters (M), while FLOPs are the numbers of FLoating point OPerations (G). The number of input table's text segment bounding boxes is 42.

where $\mathbf{P}_i$ is the vector of attention weights assigned by one head to i-th node in the graph, and H is the Shannon entropy. The trends of attention diversity variance in different blocks for different modalities with and without CCS are all shown in Fig. 4.

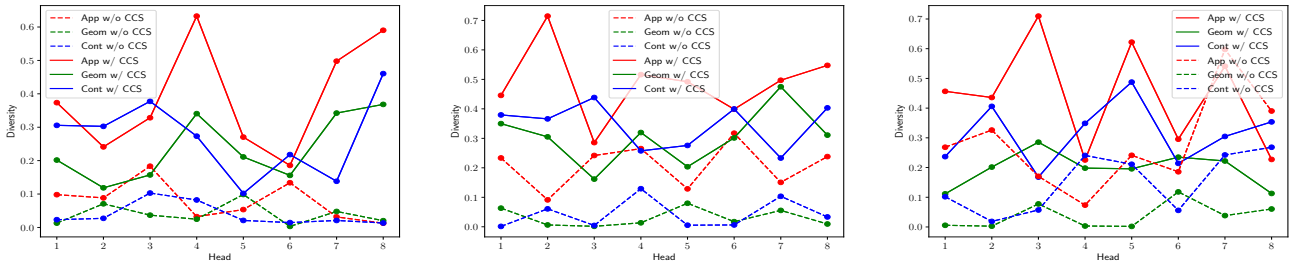
H. Qualitative Results

Fig. 5 demonstrates more qualitative results of structure recognition on benchmark datasets. The figures show the generalization ability of our proposed NCGM which is able to correctly recognize various types of table structures. Especially for more challenging cases, Fig. 5(f)-(g) verify that our method can not only handle regular tables but also robustly recognize distorted ones, which is more applicable in realistic scenarios.

We also show the failure cases of our method in Fig. 6. As one can see, the table that impairs the performance of our algorithm is the nested table, which contains severe misalignment of row and column. To put it in another way, it is ambiguous to judge whether certain boxes belong to the same row or column. The ambiguity also incurs inadaptability of existing evaluation protocols in either logical or physical format requiring the rigid alignment of box boundary in row or column relationships. In the future work, we will investigate this problem and attempt to attack it by introducing more robust representation of the nested table structure, such as tree structure.

I. Broader Impact

Table elements have natural graph structure. Learning collaborative patterns from graph data of multiple modalities offers many potential applications and opportunities as graph data in multiple modalities naturally co-occur and have implicit relationships. Our model can be applied in many specific verticals ranging from financial area to medical area including large-scale heterogeneous table data, such as financial documents, medical examination reports and *etc*. And we focus on the impact our model might have



(a) Block1 (b) Block2 (c) Block3

Figure 4. Diversities of attention maps for different modalities with or without CCS in different blocks.

on them. A model that is capable of dealing with large-scale multi-modality data is extremely significant for table information registration and data analysis. With the development of smart phones, a large amount of table images are captured by mobile cameras in realistic application. Different from regular table images obtained by scanner or parsing PDF metadata, those captured by mobile device contain more distractors (*e.g.,* distortion). Table structure recognition (TSR) algorithm plays as the front-end role that converts input table image to machine readable data, which is vital to the whole document processing system. However, most of existing TSR methods are merely designed for regular tables and cannot generate satisfactory results from table cases with more challenging distractors. Thanks to the more effective capture of inter-intra modality interaction, our model tailored for Hetero-TSR can yield more precise results, especially under more challenging scenarios, which is demonstrated by extensive experiments. In other words, our model can not only greatly save labor costs and improve document processing efficiency, but show more extensibility in application scenarios. Besides, we provide a successful attempt in the direction of investigating the collaborative patterns with and between modalities. We encourage researchers to build graph embedding models based on NCGM for other graph-based tasks we can expect to be particularly beneficial.

References

[1] Zewen Chi, Heyan Huang, Heng-Da Xu, Houjin Yu, Wanxuan Yin, and Xian-Ling Mao. Complicated table structure recognition. *arXiv preprint arXiv:1908.04729*, 2019. 3, 4, 5

[2] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017. 1

[3] Gonçalo M Correia, Vlad Niculae, and André FT Martins. Adaptively sparse transformers. *arXiv preprint arXiv:1909.00015*, 2019. 5

[4] Liangcai Gao, Yilun Huang, Hervé Déjean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Lang. Icdar 2019 competition on table detection and recognition (ctdar). In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1510–1515. IEEE, 2019. 3, 4

[5] Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. Icdar 2013 table competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1449–1453. IEEE, 2013. 3, 4

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[8] Kenneth I Joy. Quadratic bezier curves. *Department of Computer Science, University of California,, Davis*, pages 1–6, 2000. 4

[9] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Tablebank: Table benchmark for image-based table detection and recognition. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1918–1925, 2020. 3

[10] Hao Liu, Xin Li, Bing Liu, Deqiang Jiang, Yinsong Liu, Bo Ren, and Rongrong Ji. Show, read and reason: Table structure recognition with flexible context aggregator. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1084–1092, 2021. 4, 5

[11] Rujiao Long, Wen Wang, Nan Xue, Feiyu Gao, Zhibo Yang, Yongpan Wang, and Gui-Song Xia. Parsing table structures in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 944–952, 2021. 3, 4, 5

[12] Shah Rukh Qasim, Hassan Mahmood, and Faisal Shafait. Rethinking table recognition using graph neural networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 142–147. IEEE, 2019. 2, 5

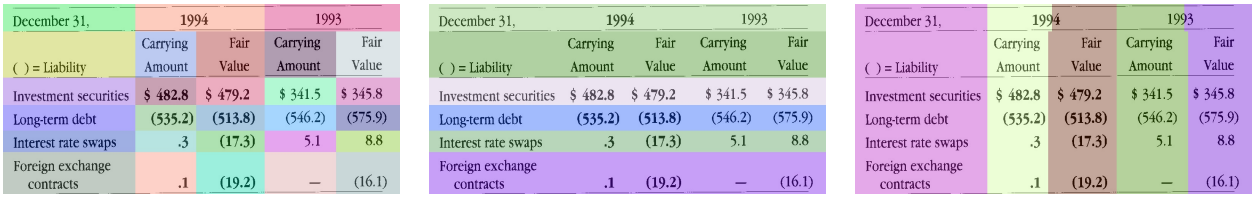
[13] Liang Qiao, Zaisheng Li, Zhanzhan Cheng, Peng Zhang, Shiliang Pu, Yi Niu, Wenqi Ren, Wenming Tan, and Fei Wu. Lgpma: Complicated table structure recognition with local and global pyramid mask alignment. *arXiv preprint arXiv:2105.06224*, 2021. 5

[14] Sachin Raja, Ajoy Mondal, and CV Jawahar. Table structure recognition using top-down and bottom-up cues. In *European Conference on Computer Vision*, pages 70–86. Springer, 2020. 3, 5

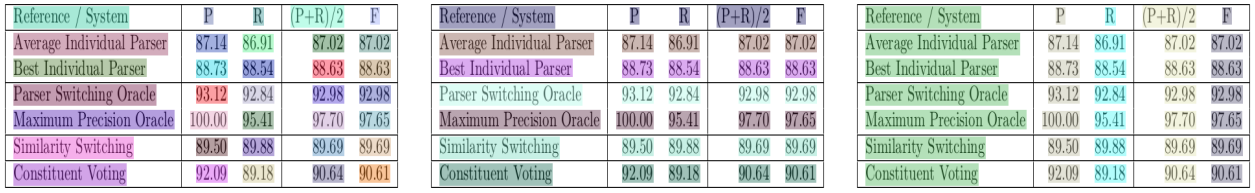
[15] Asif Shahab, Faisal Shafait, Thomas Kieninger, and Andreas Dengel. An open approach towards the benchmarking of table structure recognition systems. In *Proceedings of the 9th*

(a) Sample result of NCGM on ICDAR-2013 dataset.

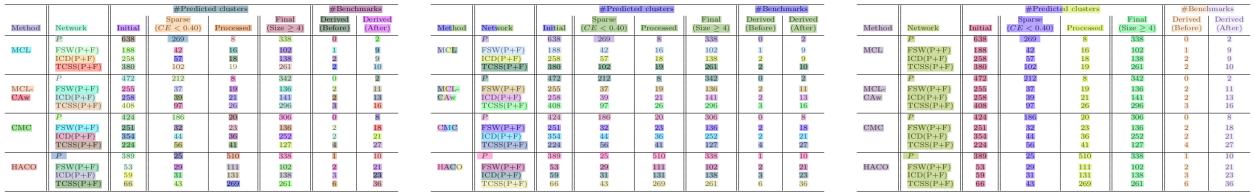
(b) Sample result of NCGM on ICDAR-2019 dataset.



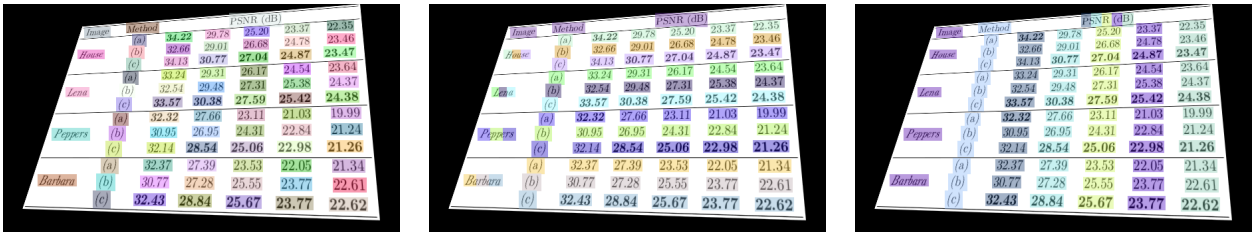
(c) Sample result of NCGM on UNLV dataset.



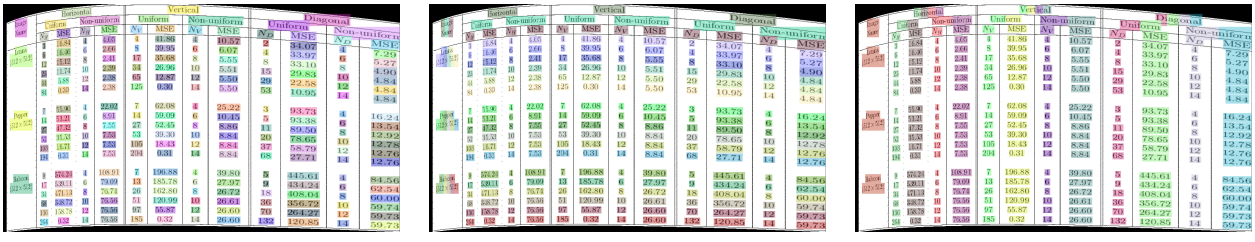
(d) Sample result of NCGM on SciTSR dataset.



(e) Sample result of NCGM on SciTSR-COMP dataset.

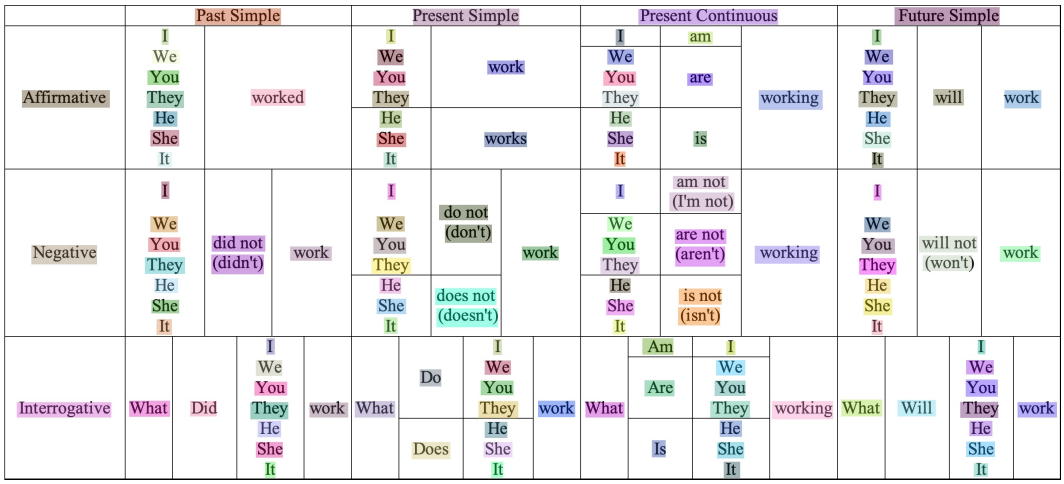


(f) Sample result of NCGM on SciTSR-COMP-A (Distortion 1) dataset.

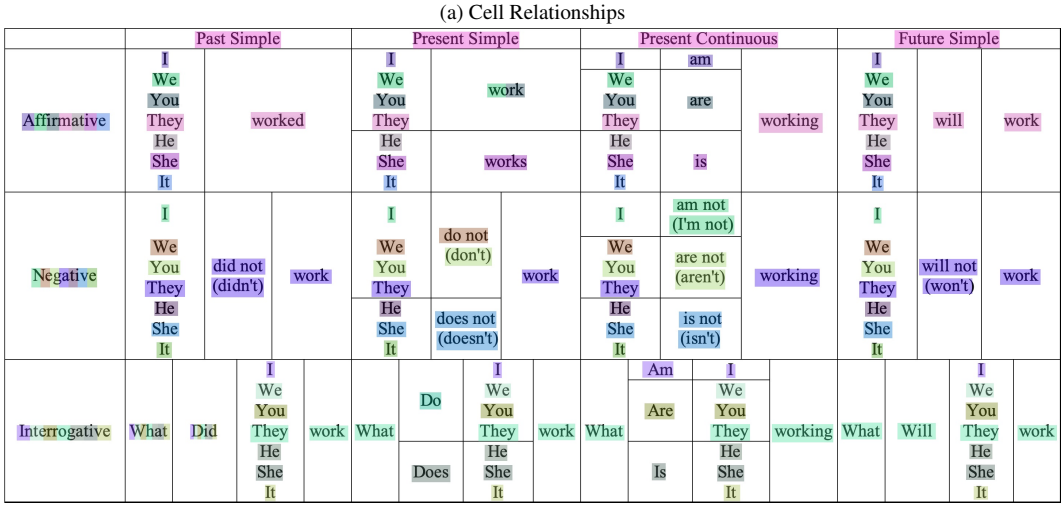


(g) Sample result of NCGM on SciTSR-COMP-A (Distortion 2) dataset.

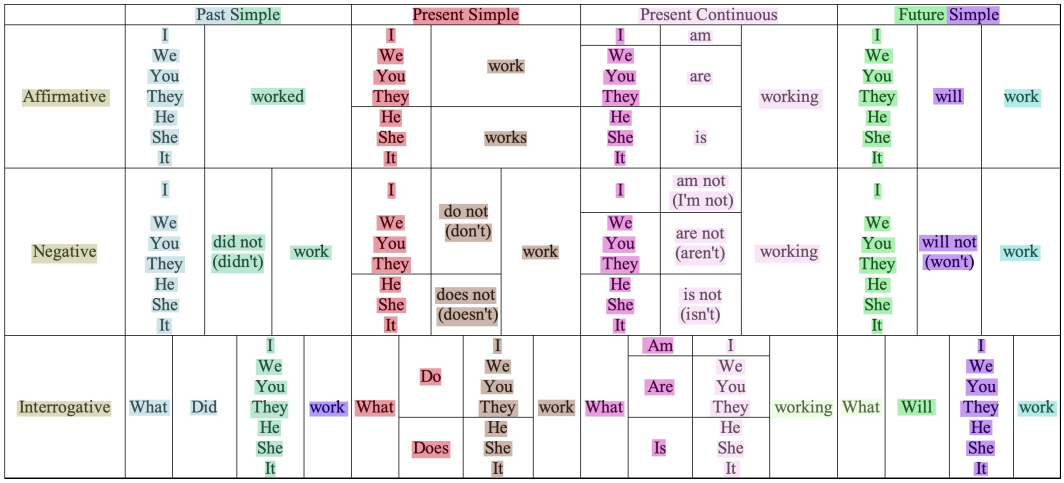
Figure 5. Sample TSR output of NCGM on table images of various datasets. The first, second and last column indicate the predictions of cells, rows and columns respectively.



(a) Cell Relationships



(b) Row Relationships



(c) Column Relationships

Figure 6. Failure cases of NCGM on table with more complex structure.

IAPR International Workshop on Document Analysis Systems, pages 113–120, 2010. 3, 4

[16] Ray Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recogni-*

tion (ICDAR 2007), volume 2, pages 629–633. IEEE, 2007. 4

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017. 1

[18] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: data, model, and evaluation. *arXiv preprint arXiv:1911.10683*, 2019. 3