LRATNet: Local-Relationship-Aware Transformer Network for Table Structure Recognition

Guangjie Yang[1], Dajian Zhong[2], Yu-jie Xiong[3], and Hongjian Zhan[1,2](✉)

[1] Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai, China
51215904041@stu.ecnu.edu.cn, ecnuhjzhan@foxmail.com

[2] College of Information Engineering, Shanghai Maritime University, Shanghai, China
djzhong@shmtu.edu.cn

[3] Shanghai University of Engineering Science, Shanghai, China
xiong@sues.edu.cn

Abstract. Table structure recognition is a challenging task due to complex background and various styles of tables. Existing methods address this challenge by exploring adjacency relationship prediction, image-to-text generation, logical position prediction, etc. However, these methods either adopt Graph Convolutional Network (GCN) structures, which mainly focus on the local context information, or Multi-Head Attention (MHA) structures, which mainly focus on the global context information. Both of them ignore the correlation between local and global features. In this paper, we propose a Local-Relationship-Aware Transformer Network (LRATNet) for table structure recognition. LRATNet constructs a robust correlation between local and global information using the LRAT module. The LRAT model has been adapted into three distinct variants: Row-LRAT, Col-LRAT, and Spa-LRAT. These variants are designed to emphasize specific aspects of information: row information, column information, and spatial information, respectively. This is achieved through the exploration of different adjacency relationships. This improves the performance of logical location prediction. Additionally, we have developed a new loss function called Lstage, which is designed to improve accuracy in predicting logical positions. Experimental results demonstrate that our method outperforms existing approaches on three public datasets.

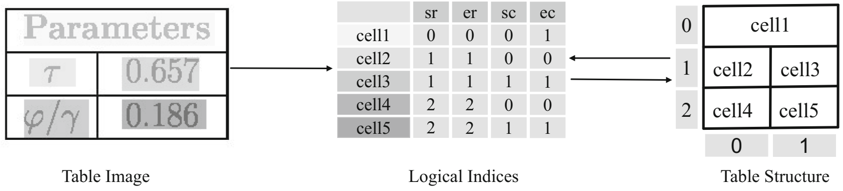
Keywords: Table structure recognition · GCN · Transformer · Adjacency encoding

1 Introduction

Table structure recognition (TSR) is an important task in the field of computer vision, aiming to automatically understanding and processing the structure and content of tables in images or documents. This facilitates various applications, including information extraction, document automation, database population,

S. Rudinac et al. (Eds.): MMM 2024, LNCS 14555, pp. 507–520, 2024.
https://doi.org/10.1007/978-3-031-53308-2_37

question answering and dialogue systems. Although many relevant works have been proposed and they have achieved impressive progress in addressing the TSR problem. It remains a challenging task due to its complex table styles, such as merged cells, spanning rows and columns, dense cell layouts, various table layouts, table images distortion and etc. Some methods [1,2,11–13,16,17] involve predicting relationships between table cells, specifically determining whether adjacent cells are in the same row or column. However, these approaches do not provide direct solutions for determining the logical positions of individual cells within tables. As a result, they often necessitate the use of complex post-processing or graph optimization algorithms to obtain the desired structural information. Some methods, as discussed in [3,7,10,25] involve modeling the TSR problem as a sequence prediction task. In this task, the table structure is predicted using corresponding LaTeX or HTML tags, which are then evaluated using Tree-Edit-Distance-based Similarity (TEDS) [25]. Other methods [20–22] adopt a more direct approach by predicting the logical positions of individual table cells. They aim to predict the starting row, starting column, ending row, and ending column of each cell, allowing for table reconstruction without the need for intricate post-processing, as depicted in Fig. 1. This approach offers a more intuitive way for machines to understand table structure. For instance, TGRNet [22] utilizes Graph Convolutional Networks (GCN) [9] and ordered node classification to predict the logical positions of rows and columns separately. GCNs are primarily designed for aggregating local context information. Although TGRNet incorporates global edge distance information as graph weights, the performance of GCNs in aggregating global context information remains limited. In contrast, LORE [20] employs Multi-Head Attention (MHA) blocks [19] to directly aggregate context information for regression-based predictions. Attention is renowned for its ability to capture global context information and have achieved impressive results in LORE. However, it is worth noting that LORE places relatively less emphasis on the role of local context information.



	sr	er	sc	ec
cell1	0	0	0	1
cell2	1	1	0	0
cell3	1	1	1	1
cell4	2	2	0	0
cell5	2	2	1	1

Fig. 1. Overview of logical location prediction paradigm. Here, sr, er, sc, ec refer to the starting row, ending row, starting column and ending column respectively.

In response to the existing problems for logical position prediction, we propose a novel approach called LRATNet, which is based on GCN and MHA architectures. LRATNet aims to solve the TSR challenge by aggregating both local and global context information. In the traditional MHA structure, the query (q), key (k), and value (v) are derived from a single feature and individually processed through three linear layers before entering MHA for information aggregation.

However, MHA excels at extracting global context information but falls short in addressing local context information. To address this problem, we propose a novel module named LRAT, which sends the key (k) and value (v) through a GCN for local context information aggregation. As a result, when the query (q) interact with the key (k) during searches, they now consider not only the global context information but also pay attention to the local context information. Inspired by [23], we have also introduced a novel strategy for the adjacency encoding and incorporated a convolutional layer within the LRAT structure to further emphasize the significance of local context information. According to the different ways of aggregating information, we also proposed three other modules, Row-LRAT, Col-LRAT, Spa-LRAT. They are employed to aggregate the context information of rows, columns and spaces respectively. Furthermore, an LRAT module is employed to aggregate these three types of context information. The above four modules together constitute a LRAT blcok. In addition, we have also designed a new loss function called the Lstage loss, which is primarily used to emphasize the accuracy of predicting all the correct logical indices for each cell, rather than solely focusing on the accuracy of a single logical index.

We summarize our contributions as follows:

- We proposed a TSR model for predicting the logical position of the table, named LRATNet. Multiple modules, such as Row-LRAT, Col-LRAT, Spa-LRAT, and LRAT, are employed to aggregate context information from various aspects.
- We proposed a LRAT module based on GCN and Transformer to pay attention to both global and local context information.
- We designed a loss function named Lstage to focus on predicting the accuracy of four logical indices for each cell, rather than solely focusing on the single logic index.

2 Related Work

Early works to TSR heavily relied on predefined rules and hand-crafted features [5,8,18]. However, rule-based methods often struggle to provide robust results for diverse table structures. In recent years, deep learning-based approaches have emerged to tackle TSR, demonstrating promising outcomes. These methods can be categorized into three main groups: adjacency relation prediction methods, image-to-text generation methods, and logical location prediction methods.

2.1 Adjacency Relation Prediction

The prediction of adjacency relationships aims to determine the positional relationship between two candidate cells in a table, specifically whether two cells are in the same row, the same column, or if they represent the same cell. Methods [1,2, 11,16,21] classify relationships between cells by treating cells as nodes and the connections between them as edges, and then a graph structure can be constructed.

FLAG-NET [13], NCGM [12], and TabStruct-Net [17] employ DGCNN [15] to model the positional relationships between adjacent cells. These methods utilize precise network structures or multimodal information to enhance relationship prediction, achieving impressive results. However, they primarily focus on predicting relationships between local cells. To capture the overall structural information of the entire table, complex post-processing or graph optimization algorithms are often required. Additionally, in specific scenarios, predicting only adjacent relationships may not suffice to infer the complete structure of the entire table.

2.2 Image-to-Text Generation

Image-to-text generation methods treat table structure prediction as a sequence prediction task. In this framework, the sequence represents the structural information of the table, often encoded in formats such as HTML or LaTeX. Sequence decoders are employed to generate textual label tags that describe the table's structure. Recent research within this paradigm predominantly adopts an end-to-end methods. Deng et al. [3] utilize an attention mechanism with an LSTM [6] structure to generate LaTeX labels that define the table's structure. EDD [25] and TableBank [10] employ an encoder-decoder architecture to produce corresponding sequence labels in an end-to-end manner. In addition, VAST [7] has introduced a visual-alignment loss, which maximizes the utilization of local information from table cells. However, it's important to note that these methods often encounter challenges related to learning noisy markup sequences. This can lead to difficulties in achieving training convergence and result in highly time-consuming sequential decoding processes.

2.3 Logical Location Prediction

Logical location prediction involves determining the starting and ending rows and columns of each cell in a table, providing vital structural information. TGRNet [22] was the pioneer in this field to our knowledge, utilizing a GCN [9] structure to integrate visual and positional cell data and employing ordered node classification for separate row and column predictions. While effective for simpler tables, its performance falls short for more complex structures. Conversely, LORE [20] employs cascaded MHA [19] blocks for direct cell position regression. However, its exclusive reliance on the MHA structure for global context aggregation overlooks the importance of local context information.

3 Methodology

3.1 Overall Architecture

The overall architecture of our proposed LRATNet model is illustrated in Fig. 2. It primarily consists three main components: a CNN-based visual feature extractor, three LRAT blocks, and a Regression module. Firstly, we use a feature

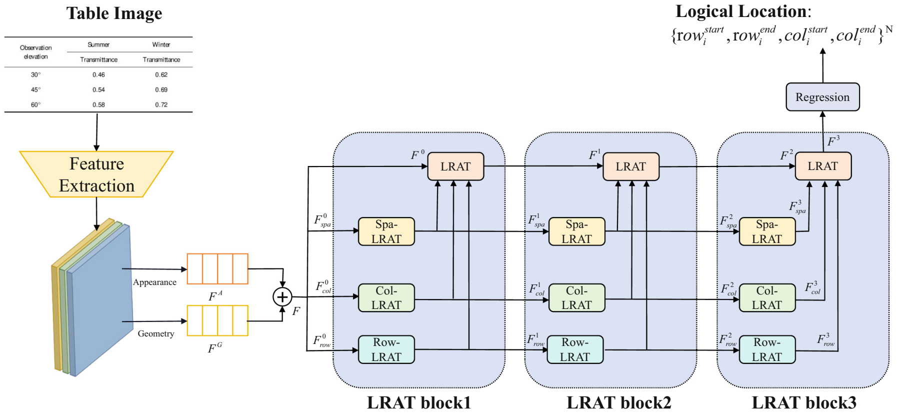


Fig. 2. The architecture of our proposed LRATNet.

extractor to acquire the visual appearance features F^A and geometric features F^G of the table cells. Then, these two sets of features are fused and fed into multiple stacked LRAT blocks. Finally, the aggregated context information is fed into the Regression Module, which predicts the logical positions of each table cell, including the starting row, ending row, starting column, and ending column.

3.2 Feature Extraction

Inspired by [14,26], for a given input table image, we employ a key point segmentation network to extract features from the table image. Based on the ground truth for each cell, we model the visual features of each cell as the sum of its four corner points, denoted as F^{cr}, and the center point, denoted as F^{ct}. $F^{cr} = \left\{f^1_{cr}, f^2_{cr}, ..., f^N_{cr}\right\} \subset R^{N\times d}$, where N and d represent the number of each table cells and the embedding dimension, respectively. $F^{ct} = \left\{f^1_{ct}, f^2_{ct}, ..., f^N_{ct}\right\} \subset R^{N\times d}$. Here, f^i_{cr} corresponds to the features on the feature map of the bounding box corners, f^i_{ct} corresponds to the features on the feature map of the center point, and i denotes the i-th cell. Therefore, the visual embedding for the table image is denoted as:

$$F^A = F_{ct} + F_{cr}. \tag{1}$$

The geometric embedding, denoted as $F^G = \left\{f^1_b, f^2_b, ..., f^N_b\right\}$, is formed by concatenating the corner points and the center point of each cell. These embeddings are obtained using a Fully-Connected layer for embedding. Subsequently, we obtain the final feature representation $F = F^A + F^G = \left\{f^1, f^2, ...f^N\right\} \subset R^{N\times d}$.

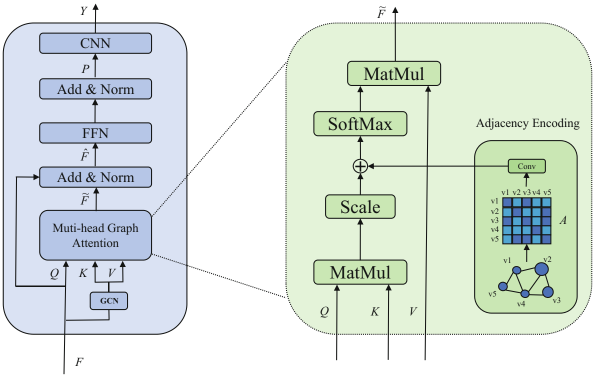


Fig. 3. The internal structure of our proposed LRAT module. A is the adjacency matrix.

3.3 LRAT Module

In this section, we will provide a detailed overview of the LRAT module. As shown in Fig. 3. The LRAT module mainly consists of convolutional layers, linear layers, a GCN layer, and multi-head graph attention components. For the input F, and Q, K and V are first calculated when the feature F is fed to the LRAT module. Next, we use F_Q, F_K and F_V to respectively represent the features calculated for Q, K and V. For F_Q, $F_Q = F$. Regarding K and V, to improve aggregation of local context information, and to facilitate enhanced focus on both local and global context information when querying K with Q later on, we employ GCN [9] to aggregate local context information. Which are denoted as:

$$F_V = F_K = GCN(F, A). \tag{2}$$

Secondly, we utilize Multi-Head Graph Attention (MGHA) to aggregate global context information. Following the MGHA operation, we obtain $\tilde{F} = MGHA(F_Q, F_K, F_V)$. Similar to the approach presented in [19], we apply residual connections and layer normalization to the LRAT module. After obtaining P, we introduce an additional CNN layer to further aggregate adjacent information at the end.

$$\hat{F} = Add\&Norm(\tilde{F}, F_Q), \tag{3}$$

$$P = Add\&Norm(FFN(\hat{F}), \hat{F}), \tag{4}$$

$$Y = CNN(P). \tag{5}$$

For the MGHA operation, drawing inspiration from prior work [23], we start by passing Q and K through a MatMul layer followed by a Scale layer. Additionally, we introduce a 1×1 convolution operation on the adjacency matrix to encode it into the same dimension as K and V. Subsequently, we sum these components together and apply a SoftMax function. This process further enhances the emphasis on local context information. Finally, $\tilde{F}$ can be obtained through a MatMul layer.

3.4 LRAT Block

In Sect. 3.3, we provided a detailed overview of the LRAT module. In this section, we introduce our proposed LRAT block. The LRAT block consists of four modules: Row-LRAT, Col-LRAT, Spa-LRAT, and LRAT. The structures of these four modules are the same as the LRAT module introduced in Sect. 3.3. However, we distinguish them here primarily based on how they aggregate context information. For Row-LRAT and Col-LRAT, their primary focus is on aggregating context information along rows and columns, respectively. To achieve this, when constructing the adjacency matrix for GCN input, we use Row-LRAT to exclusively capture row adjacency relationships in an adjacency matrix, while employing Col-LRAT to exclusively address column adjacency relationships in an adjacency matrix. This approach follows a similar methodology as described in [1]. In our modeling, each cell is considered a node, and we represent its edges as e_{ij} (where i is not equal to j, and 1 <= i, j <= N). Consequently, when constructing edges, we calculate edges between each node and all other nodes, utilizing the Euclidean distance as the corresponding weight for each edge. The specific weight calculation is as follows:

$$A_{i,j}^{row} = exp\{-(\frac{b_i^y - b_j^y}{H} \times \beta)^2\}, \tag{6}$$

$$A_{i,j}^{col} = exp\{-(\frac{b_i^x - b_j^x}{W} \times \beta)^2\}, \tag{7}$$

Here, $A_{i,j}^{row}$ represents the row relationship, $A_{i,j}^{col}$ represents the column relationship, b_i^x and b_i^y correspond to the x and y coordinates of each cell's center point, and H and W denote the width and height of the image, respectively. β is a hyperparameter. In this process, we focus solely on one-dimensional location relationships within rows or columns, rather than spatial relationships. In contrast to the previous two modules, Spa-LRAT primarily aggregates information based on adjacency relationships along the row and column dimensions, which mainly focuses on spatial location relationships. The structure of Spa-LRAT is identical to that of LRAT, and the final information aggregation is performed by LRAT. The overall feature representation is defined as $F^c = F_{row}^c + F_{col}^c + F_{spa}^c$. Here, F_{row}^c, F_{col}^c, and F_{spa}^c represent the features after aggregation by Row-LRAT, Col-LRAT, and Spa-LRAT at a specific layer denoted by c.

3.5 Regression Module

After aggregating local and global context information through three LRAT blocks, we obtain the final context information. Subsequently, we employ a Multi-Layer Perceptron (MLP) for the regression task, predicting the starting row, ending row, starting column, and ending column for each cell. This process produces the ultimate results, denoted as $L = \left\{l^1, l^2, ..., l^N\right\}$, where $l^i = \left\{row_i^{start}, row_i^{end}, col_i^{start}, col_i^{end}\right\}$.

3.6 Loss Function

Our loss function is computed as $L_{all} = aL_{log} + bL_{stage}$, where a and b are hyperparameters. It comprises two main components: L_{log} and L_{stage}. L_{log} quantifies the difference between the predictions directly generated by the regression module and the corresponding ground truth values. Regarding L_{stage}, we only consider the logical position of a cell as correct when all four logical indices are accurately predicted. Even if three out of four indices are correct, it is still considered an incorrect prediction. To address this, we have devised a multi-stage loss function called L_{stage}. Specifically, l_c^i is calculated as the number of correctly predicted logical indices based on l^i for each cell. $\tilde{l^i}$ represents the penalty value we have designed based on different l_c^i values. $\hat{l^i}$ represents the ground truth. The specific calculation formula is as follows:

$$L_{log} = \frac{1}{N}\sum_{i=1}^{N} ||l^i - \hat{l^i}||_1, \tag{8}$$

$$\tilde{l^i} = \begin{cases} 2.5\ , & l_c^i = 0 \\ 2\ , & l_c^i = 1 \\ 1.5\ , & l_c^i = 2 \\ 1\ , & l_c^i = 3 \\ 0\ , & l_c^i = 4 \end{cases} \tag{9}$$

$$L_{stage} = \frac{1}{N}\sum_{i=1}^{N} \tilde{l^i}. \tag{10}$$

4 Experiments

In this section, we evaluate the performance of our proposed LRATNet on three datasets and compare it with existing methods for predicting table logical positions. Additionally, we conduct comprehensive ablation experiments to validate the effectiveness of LRATNet.

4.1 Datasets

We evaluate the performance of LRATNet on three datasets related to table structure recognition, namely WTW [14], TableGraph-24K [22], and ICDAR2013 [4]. Due to the limited size of the ICDAR2013 dataset, we adopted a fine-tuning approach following initial training on the TableGraph-24K dataset.

WTW [14]: This dataset comprises wired tables found in natural scenes and spans across multiple categories, making it a significant challenge. It includes 10,037 training images and 3,600 testing images.

TableGraph-24K [22]: This dataset is a subset of TableGraph-350K, consisting of 20,000 training images and 2,000 testing images. It contains both wired and wireless textual tables.

ICDAR2013 [4]: This dataset consists of 156 PDF format tables from EU/US government websites. However, since it lacks a dedicated testing set, we have adopted the data processing approach used in TGRNet [22] to ensure a fair comparison. We randomly divided the dataset into two halves, with one half serving as the training set and the other as the testing set.

4.2 Implementation Details

All experiments were conducted using an RTX 3090 GPU with CUDA 11.3 and PyTorch 1.11. The labels for the ICDAR2013 and TableGraph-24K (TG24K) datasets were transformed following the TGRNet repository, whereas the labels for the WTW dataset were transformed based on the LORE repository. For TG24K and ICDAR13, we proportionally scaled the images to 768×768 and trained the proposed network for 200 epochs. The results for ICDAR2013 were obtained by fine-tuning the model pre-trained on TG24K. As for the WTW dataset, we proportionally scaled the images to 1024×1024 and trained the model for 100 epochs with a batch size of 6. We employed the DLA-34 [24] as the backbone with a dimension d set to 256. Additionally, we incorporated 3 layers of LRAT blocks. The initial learning rate was set to 0.0001 and was reduced by 10% at the 70th and 90th epochs. Hyperparameters were configured as follows: $a = 2$, $b = 1$, and $\beta = 3$.

4.3 Results on Benchmarks

We compared our proposed LRATNet with existing methods for table logical location prediction, including ReS2TIM [21], TGRNet [22], and LORE [20]. While ReS2TIM is similar to ours in that it directly uses the ground truth of the table cell as input, both TGRNet and LORE are end-to-end methods, which may not achieve complete accuracy in table cell detection. To ensure a fair comparison, we made slight modifications to their code. Instead of using the partially accurate detected table cells for logical position prediction, we provided the ground truth of the table cell information before the phase of logical position prediction. The experimental results are summarized in Table 1.

Table 1. Comparison results with existing methods for logical position prediction. The symbol '*' denotes methods that have been reproduced.

Method	ICDAR13	TG24K	WTW
ReS2TIM [21]	17.4	-	-
TGRNet* [22]	33.4	91.4	-
LORE* [20]	75.2	96.5	85.0
Ours	79.5	98.0	89.6

Our proposed LRATNet pays attention to both local and global context information. Experimental results demonstrate that our approach outperforms

ReS2TIM, TGRNet, and LORE on three datasets, highlighting the effectiveness of our model in considering both local and global context information. The ReS2TIM primarily relies on a cell relationship network, but its ability to capture both local and global context information is limited, resulting in poor performance. The TGRNet employs a GCN and uses global edge connections as weights, which address both local and global context information to some extent. However, when compared to a Transformer-based architecture [20], the capacity of GCN for aggregating context information is limited. Consequently, TGRNet performs better than ReS2TIM but falls short of LORE. LORE adopts a Transformer-based structure and achieves commendable results. However, it overlooks the significance of local context information.

The relatively high accuracy of the three methods for TG24K can be attributed to the fact that this dataset primarily consists of scientific literature, features relatively simple structures, and benefits from ample training data. However, the results of ICDAR2013, fine-tuned from models pretrained on the TG24K, did not match the high performance of the TG24K. We speculate that this discrepancy may arise from the limited size of the training dataset for the ICDAR2013, hindering the model of ability to effectively capture its structural characteristics. For the WTW dataset, known for its intricate and diverse structures, it presented a significant challenge. However, our model achieved an impressive accuracy rate of 89.6%, highlighting its robust performance.

Considering that our method relies on the ground truth of table cells, it might not always be possible to achieve perfectly accurate table cell detection results in real situations. To validate the effectiveness of our approach, given that the current accuracy of most table cell detection methods exceeds 80%, we randomly reduced the number of cells by 10% and 20% for training our model. The results are shown in Table 2. From the experimental results, it can be observed that our proposed LRATNet maintains strong robustness even on incomplete table structures.

Table 2. Robustness analysis results on the WTW dataset.

Method	100% cells	90% cells	80% cells
LRATNet	89.6	87.0	85.3

4.4 Ablation Study

In this section, we have performed a comprehensive set of ablation experiments on the WTW dataset to evaluate the effectiveness of our proposed LRATNet. The specific experiments are outlined as follows:

Analysis of the Effects of Our Proposed Row-LRAT, Col-LRAT, Spa-LRAT, and LRAT. Our experimental results are presented in Table 3. From

Table 3. Evaluation the significance of Row-LRAT, Col-LRAT, Spa-LRAT, and LRAT modules. Acc represents the accuracy of predicting the logical position of each cell.

Exp	Row-LRAT	Col-LRAT	Spa-LRAT	LRAT	Acc
1			✓		88.4
2	✓	✓			87.7
3	✓	✓	✓		88.5
4	✓	✓		✓	89.1
5	✓	✓	✓	✓	89.6

the Exp. 1–2 and Exp. 4, it can be observed that our proposed LRAT and Spa-LRAT have played a crucial role in improving the performance. This can be attributed to their ability to focus on both local and global context information. From Exp. 1 and Exp. 2, it is evident that Row-LRAT and Col-LRAT do not perform as well as the single Spa-LRAT. This is because they separately aggregate context information from rows and columns, lacking spatial context information. However, from Exp. 2 and Exp. 4, we observe that their performance improves significantly after aggregation through LRAT. Therefore, it is evident that each module we proposed plays a crucial role in enhancing the performance.

Analysis of the Internal Component Functions in LRAT. The experimental results are presented in Table 4. In Exp. 1, only a 3-layer MHA block was used, while in Exp. 2–5, GCN, Adjacency Encoding, CNN, and Lstage loss were progressively incorporated. The experimental results indicate that the introduction of GCN and Adjacency Encoding leads to noticeable performance improvements, further emphasizing the importance of considering both global and local context information. The influence of the Lstage loss on performance is moderate. Our analysis indicates that the primary contributions to improvements come from the L1 loss, while the Lstage loss plays a supplementary role in enhancing the accuracy of the model in predicting logical positions.

Table 4. Effectiveness of the LRAT in the proposed method.

Exp	GCN	Adjacency Encoding	CNN	Lstage	Acc
1					84.1
2	✓				88.0
3	✓	✓			89.0
4	✓	✓	✓		89.3
5	✓	✓	✓	✓	89.6

Analysis of Weight Parameter Sharing for CNN and GCN Within Each LRAT Module. The experimental results are detailed in Table 5. The experimental findings demonstrate an accuracy improvement of 0.2% points whether using CNN or GCN. Furthermore, this approach results in a reduction in the total number of model parameters. Our analysis indicates that this improvement can be attributed to the roles of CNN and GCN, both of which primarily focus on aggregating local information. Given that different layers of the Multi-Head Graph Attention (MGHA) module already perform information aggregation at various depths, there is no need to further increase the depth of CNN and GCN layers. A single layer suffices for effective local information aggregation. Consequently, sharing these parameters within each LRAT module proves to be a more effective strategy.

Table 5. Results of weight parameter sharing for CNN or GCN within each type of LRAT module.

Exp	GCN	GCN (shared)	CNN	CNN (shared)	Acc
1		✓	✓		89.4
2	✓			✓	89.4
3		✓		✓	89.6

Analysis of LRATNet with Different Number of LRAT Blocks. We conducted LRATNet experiments with two-layer, three-layer, and four-layer LRAT blocks. The experimental outcomes are presented in Table 6. The results indicate that the three-layer LRAT block exhibits a significant improvement compared to the two-layer experimental results, while the four-layer LRAT block does not yield a substantial improvement in performance. Taking into consideration both model complexity and performance, we ultimately opted for a three-layer LRAT block structure.

Table 6. The results of LRATNet with varying number of LRAT blocks.

Exp	two-layers	three-layer	four-layer	Acc
1	✓			88.9
2		✓		89.6
3			✓	89.9

5 Conclusion

In this paper, We proposed LRATNet, a TSR model designed for predicting table logical positions. LRATNet incorporates four modules: Row-LRAT, Col-LRAT, Spa-LRAT, and LRAT, each individually aggregating information from rows, columns, spaces, and the overall table. This approach maximizes the utilization of both local and global context information for modeling. LRATNet also utilizes a multi-stage loss function to emphasize the accuracy of the four logical indices for each cell. Experimental results demonstrate that our approach outperforms existing methods on three public datasets.

References

1. Chi, Z., Huang, H., Xu, H.D., Yu, H., Yin, W., Mao, X.L.: Complicated table structure recognition. arXiv preprint arXiv:1908.04729 (2019)
2. Clinchant, S., Déjean, H., Meunier, J.L., Lang, E.M., Kleber, F.: Comparing machine learning approaches for table recognition in historical register books. In: IAPR International Workshop on Document Analysis Systems (DAS), pp. 133–138 (2018)
3. Deng, Y., Rosenberg, D., Mann, G.: Challenges in end-to-end neural scientific table recognition. In: International Conference on Document Analysis and Recognition (ICDAR), pp. 894–901 (2019)
4. Göbel, M., Hassan, T., Oro, E., Orsi, G.: ICDAR 2013 table competition. In: International Conference on Document Analysis and Recognition (ICDAR), pp. 1449–1453 (2013)
5. Hirayama, Y.: A method for table structure analysis using DP matching. In: International Conference on Document Analysis and Recognition (ICDAR), pp. 583–586 (1995)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Huang, Y., et al.: Improving table structure recognition with visual-alignment sequential coordinate modeling. In: International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11134–11143 (2023)
8. Kieninger, T., Dengel, A.: The t-recs table recognition and analysis system. In: Lee, S.-W., Nakano, Y. (eds.) DAS 1998. LNCS, vol. 1655, pp. 255–270. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48172-9_21
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
10. Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., Li, Z.: TableBank: table benchmark for image-based table detection and recognition. In: Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 1918–1925 (2020)
11. Li, Y., Huang, Z., Yan, J., Zhou, Y., Ye, F., Liu, X.: GFTE: graph-based financial table extraction. In: Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G.M., Mei, T., Bertini, M., Escalante, H.J., Vezzani, R. (eds.) ICPR 2021, Part II. LNCS, vol. 12662, pp. 644–658. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68790-8_50
12. Liu, H., Li, X., Liu, B., Jiang, D., Liu, Y., Ren, B.: Neural collaborative graph machines for table structure recognition. In: International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4533–4542 (2022)
13. Liu, H., et al.: Show, read and reason: table structure recognition with flexible context aggregator. In: ACM International Conference on Multimedia (ACM MM), pp. 1084–1092 (2021)
14. Long, R., et al.: Parsing table structures in the wild. In: International Conference on Computer Vision (ICCV), pp. 944–952 (2021)

15. Qasim, S.R., Kieseler, J., Iiyama, Y., Pierini, M.: Learning representations of irregular particle-detector geometry with distance-weighted graph networks. Eur. Phys. J. C **79**(7), 1–11 (2019)
16. Qasim, S.R., Mahmood, H., Shafait, F.: Rethinking table recognition using graph neural networks. In: International Conference on Document Analysis and Recognition (ICDAR), pp. 142–147 (2019)
17. Raja, S., Mondal, A., Jawahar, C.V.: Table structure recognition using top-down and bottom-up cues. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12373, pp. 70–86. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58604-1_5
18. Tupaj, S., Shi, Z., Chang, C.H., Alam, H.: Extracting tabular information from text files. EECS Department, Tufts University, Medford, USA (1996)
19. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
20. Xing, H., et al.: LORE: logical location regression network for table structure recognition. In: Association for the Advancement of Artificial Intelligence Conference (AAAI), pp. 2992–3000 (2023)
21. Xue, W., Li, Q., Tao, D.: ReS2TIM: reconstruct syntactic structures from table images. In: International Conference on Document Analysis and Recognition (ICDAR), pp. 749–755 (2019)
22. Xue, W., Yu, B., Wang, W., Tao, D., Li, Q.: TGRNet: a table graph reconstruction network for table structure recognition. In: International Conference on Computer Vision (ICCV), pp. 1295–1304 (2021)
23. Ying, C., et al.: Do transformers really perform bad for graph representation? arXiv preprint arXiv:2106.05234 (2019)
24. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2403–2412 (2018)
25. Zhong, X., ShafieiBavani, E., Jimeno Yepes, A.: Image-based table recognition: data, model, and evaluation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12366, pp. 564–580. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_34
26. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)