Grab What You Need: Rethinking Complex Table Structure Recognition with Flexible Components Deliberation

Hao Liu[†*], Xin Li[*], Mingming Gong, Bing Liu, Yunfei Wu, Deqiang Jiang, Yinsong Liu, Xing Sun

Tencent YouTu Lab

hfut.haoliu@gmail.com, {fujikoli, riemanngong, billbliu, marcowu, dqiangjiang, jasonysliu, winfredsun}@tencent.com

Abstract

Recently, Table Structure Recognition (TSR) task, aiming at identifying table structure into machine readable formats, has received increasing interest in the community. While impressive success, most single table component-based methods can not perform well on unregularized table cases distracted by not only complicated inner structure but also exterior capture distortion. In this paper, we raise it as Complex TSR problem, where the performance degeneration of existing methods is attributable to their inefficient component usage and redundant post-processing. To mitigate it, we shift our perspective from table component extraction towards the efficient multiple components leverage, which awaits further exploration in the field. Specifically, we propose a seminal method, termed GrabTab, equipped with newly proposed Component Deliberator, to handle various types of tables in a unified framework. Thanks to its progressive deliberation mechanism, our GrabTab can flexibly accommodate to most complex tables with reasonable components selected but without complicated post-processing involved. Quantitative experimental results on public benchmarks demonstrate that our method significantly outperforms the state-of-the-arts, especially under more challenging scenes.

Introduction

With the fast-paced development of digital transformation, Table Structure Recognition (TSR) task, aiming at parsing table structure from a table image into machine-interpretable formats, which are often presented by both table cell physical coordinates (Schreiber et al. 2017; Paliwal et al. 2019; Khan et al. 2019; Tensmeyer et al. 2019; Chi et al. 2019; Qasim, Mahmood, and Shafait 2019; Raja, Mondal, and Jawahar 2020; Zheng et al. 2021; Qiao et al. 2021; Liu et al. 2021; Long et al. 2021) and their logical relationships (Li et al. 2020; Zhong, ShafieiBavani, and Yepes 2019). It has received increasing research interest due to the vital role in many document understanding applications (Jauhar, Turney, and Hovy 2016; Li et al. 2016; Feng et al. 2023a,b). To date, several pioneers works (Li et al. 2020; Zhong, ShafieiBavani, and Yepes 2019; Schreiber et al. 2017; Khan et al. 2019; Tensmeyer et al. 2019; Chi et al. 2019; Qasim, Mahmood, and Shafait 2019; Raja, Mondal, and Jawahar 2020; Zheng et al. 2021; Qiao et al. 2021; Liu et al. 2021; Long et al. 2021) have achieved significant progress in the filed, which can be mainly categorized into *boundary extraction-based methods* (Schreiber et al. 2017; Paliwal et al. 2019; Khan et al. 2019; Tensmeyer et al. 2019; Long et al. 2021; Ma et al. 2023) and *element relationship-based methods* (Liu et al. 2021; Chi et al. 2019; Qasim, Mahmood, and Shafait 2019; Raja, Mondal, and Jawahar 2020) according to the component type leveraged.

Unfortunately, the above single-component-based tech-

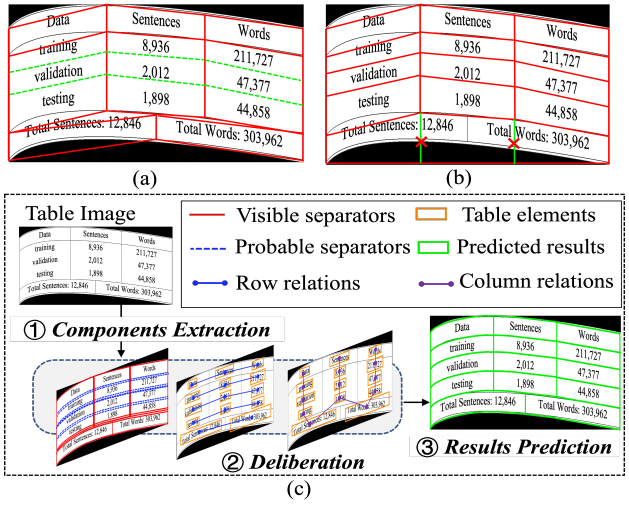

Figure 1: Illustration of motivation of the proposed GrabTab. (a) Boundary extraction-based methods. (b) Element relationship-based methods. As merely one single table component leveraged, the predicted cell boundaries could suffer from "boundary missing" (green dashed lines) or "over-prediction" (green solid lines) problem. (c) Our proposed GrabTab. A set of table components, including row/column relations, table elements, visual explicit and implicit separators, are "deliberated" by our GrabTab, where informative clues are flexibly picked up and assembled, which is more versatile for various complex table layouts. Best viewed in color.

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

niques can only yield promising results on regularized table cases, but struggle when processing more complicated cases as illustrated in Fig. 1(a) and (b). However, with the popularization of mobile capture devices, the requirement on recognizing camera-captured tables has become increasingly imperative. In this scene, other than complicated table inner structure, geometrical distortion incurred by the capturing process becomes another distracting factor. In this paper, we define this more challenging TSR as *Complex TSR task*. We attribute the performance degeneration of previous methods to their inefficient component usage and heavy dependence on rule-based post-processing. To be specific, given a complex table, boundary-based methods can better handle the visible boundary cells by directly predicting them, but suffer from "boundary missing" problem for those without explicit separations (green dashed lines in Fig. 1(a)). Comparatively, the relationship-based alternatives can overcome this issue by inferring cell boundaries from the element relationships, whereas the "over-prediction" of boundaries (green solid lines in Fig. 1(b)) is witnessed as the visible cell boundary clues are totally abandoned. To relieve these shortcomings, both methods resort to well-designed post-processing rules, however, they would become uncontrollable when distortion happens. This *status quo* begs for a question: *"Is there a versatile TSR solution to leverage merits of multiple components rather than merely single one, in the light of different complex table cases?"*. A straightforward way is to directly combine multiple components. Nevertheless, it is infeasible to implement as either component in the aforementioned methods is strongly coupled with corresponding post-processing rules, which could be mutually exclusive. For the Complex TSR, a few recent researches (Long et al. 2021; Liu et al. 2022; Wang et al. 2023) have made attempts, whereas they only take further steps on more robust components (relationships or boundaries) extraction while the deployment of multiple components is still rarely explored.

In human cognitive system (Clancey 2002; Anderson 2005; Olshausen, Anderson, and Van Essen 1993), "deliberation" is one of common behaviors when human processing daily works, such as reading or analyzing table image. Specifically, a set of evident visual clues are perceived at a rough level and the final results are yielded by complementing them with implicit but necessary information after deliberation in a progressive way. Inspired by it, we in this paper introduce the "deliberation" mechanism and propose a novel method, termed **GrabTab**, tailored for Complex TSR problem, which can flexibly **Grabs** the needed information from a set of **Tab**le components and progressively assembles them to the final results, as demonstrated in Fig. 1(c).

Concretely, we first go beyond canonical straight line-based method and propose a new table Separator Perceiver (SP) based on Bézier curve, which can yield high-quality explicit (solid red lines) and implicit separator (dashed blue lines) proposals. Treating both types of separator proposals as hints, the newly designed Components Correlator (CC) "grabs" useful information from table elements and their global relationships aggregated on the separator proposals shown in Fig. 1(c). Afterward, requiring no sophisticated post-processing, our GrabTab directly predicts the logical index and "grabs" refined separators to constitute the final results through Structure Composer (SC). Serving as core submodules, SP, CC and SC comprise our Components Deliberator (CD) implementing progressive deliberation mechanism. Thanks to this mechanism and removal of complicated heuristic-based post-processing, our GrabTab exhibits prominent versatility, which can flexibly accommodate to most complex tables with reasonable components selected. Benefiting from the tailored design, our GrabTab method can achieve better performance compared to other TSR methods, especially for the complex table scenarios, as vividly validated by extensive experimental results. Conclusively, our contributions are summarized as:

- We reinspect the TSR task from the perspective of the efficient multiple components leverage, rather than single component extraction widely adopted by previous methods. To our best knowledge, we are the first to introduce deliberation mechanism and investigate its working patterns on component interaction for predicting complex table structure.
- We coin a novel and versatile method, GrabTab, tailored for Complex TSR problem, which is equipped with Components Deliberator consisting of Separator Perceiver, Component Correlator and Structure Composer, responsible for generation of high-quality separator proposals, multiple components correlation and composing refined separator into final results.
- Quantitative experimental results on public benchmarks demonstrate that our method can fully leverage components reciprocity for diversified complex table cases, without introducing extra complicated processes. Consequently, significant performance improvement is witnessed, especially under more challenging scenes.

Methodology

Overall Architecture

Intuitively, table separator is the most evident and straightforward visual clue, which is also the basic ingredients of the final output results. Based on this intuition, our GrabTab thus treats it as chief component to dynamically "grab" informative clues from other candidate components (the table elements (orange boxes) and their row/column relations (connected by blue and purple solid lines)) during deliberation. The architecture of our proposed GrabTab is designed as Fig. 2, which consist of four stages. Given a complex table image, firstly, the candidate components are extracted as element tokens (in orange color) and relation tokens (in green color). Then, the feature of table image along with relation bias is sent to the newly proposed Separator Perceiver (SP) to obtain separator tokens, which can generate a set of explicit (red solid lines in Fig. 2) and implicit separator (blue dashed lines) proposals through least squares fitting (LSF) (Weisstein 2002). Afterwards, Components Correlator (CC) correlates to the separator tokens (in blue color) with relation and element tokens to obtain the enhanced separator tokens (in purple color). In the end, Structure Composer (SC) selects the desired separators by predicting their

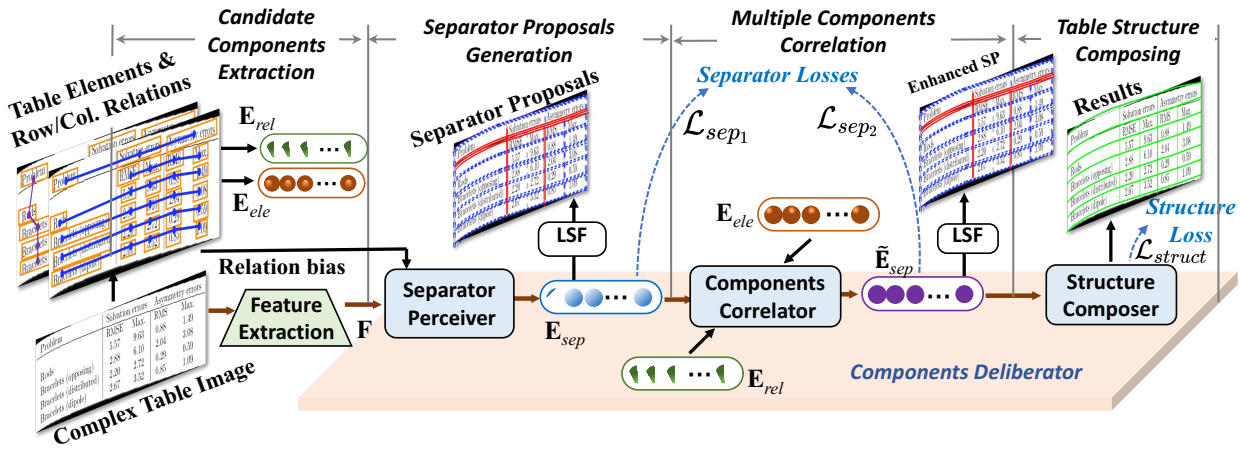

Figure 2: The architecture of our proposed GrabTab. $\mathbf{E}_{rel}$, $\mathbf{E}_{ele}$, $\mathbf{E}_{sep}$ and $\widetilde{\mathbf{E}}_{sep}$ denote relation, element, separator and enhanced separator tokens, respectively. Best viewed in color and zoomed in.

indexes in a sequential manner and re-assembles them as closure cells. The framework is end-to-end trainable by the proposed "separator losses" and "structure loss", which ensures the versatility of our GrabTab.

Candidate Components Extraction

As aforementioned, our GrabTab extracts table elements and their relations as candidate components, which is expected to provide useful information for the chief separator component. To achieve this goal, we inherit the relation extraction method from a off-the-shell work, NCGM (Liu et al. 2022). Specifically, for N table elements, the "collaborative graph embeddings" output by NCGM is employed as element tokens in our GrabTab: $\mathbf{E}_{ele} = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N\} \in \mathbb{R}^{N \times d_e}$. Correspondingly, the relation tokens $\mathbf{E}_{rel}$ are also obtained according to the binary-class relations $\{\mathbf{R}_{row}, \mathbf{R}_{col.}\}$ predicted by NCGM. In details, $\mathbf{R}_{row} = \{\mathbf{r}_{1,1}, \mathbf{r}_{1,2}, ..., \mathbf{r}_{i,j}, ..., \mathbf{r}_{N,N}\} \in \mathbb{R}^{N^2 \times 2}$, where $\mathbf{r}_{i,j} = 1$ if the pair of i-th and j-th element belong to the same row, and it equals to 0 otherwise. $\mathbf{R}_{col.}$ is denoted in the same manner. To avoid the costly computational consumption brought by element pairs in large amount, according to $\{\mathbf{R}_{row}, \mathbf{R}_{col.}\}$, we link elements with same relationship as one instance class, *i.e.*, $\mathbf{E}_{row} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_M\} \in \mathbb{R}^{M \times d_e}$, $\mathbf{E}_{col} = \{\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_P\} \in \mathbb{R}^{P \times d_e}$. Mathematically, for i-th row relation instance: $\mathbf{f}_i = \sum_{m=1}^{n} \mathbf{e}_m + \mathbf{w}_i$, where $\mathbf{w}_i \in \mathbb{R}^{d_e}$ is the i-th instance index embedding produced by method (Mikolov et al. 2013). The dictionary size is set to 200 in default. And the $\mathbf{E}_{col}$ is obtained in the similar way. Finally, the relation tokens $\mathbf{E}_{rel} \in \mathbb{R}^{((M+P) \times d_e)}$ are generated as $\mathbf{E}_{rel} = \{\mathbf{E}_{row}, \mathbf{E}_{col}\}$.

Separator Proposals Generation

Separator representation. Our aim of this stage is to obtain a set of separator proposals, including both explicit and implicit ones shown in Fig. 2, which can well cover the potential cell boundary regions. However, most of previous methods represent cell boundaries with straight lines, which could not fit them properly under distortion scene (see Fig. 1(a) and (b)). Inspired by work (Liu et al. 2020), we directly deploy the cubic Bézier curve to represent the curved separators, which is defined as:

$$\mathcal{B}(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3, t \in [0, 1],$$

where P_0, P_1, P_2 and P_3 denote control points of the Bézier curve. Intuitively, a straightforward way to produce curved separator is to directly predict the curvature control points. However, it may suffer from instable training procedure and "point drift" problem, *i.e.*, small point prediction error can lead to the whole curve misalignment. Alternatively, considering the Bézier curve is parameterized in terms of $t \in [0, 1]$, we sample the curve uniformly from t spaced set to get

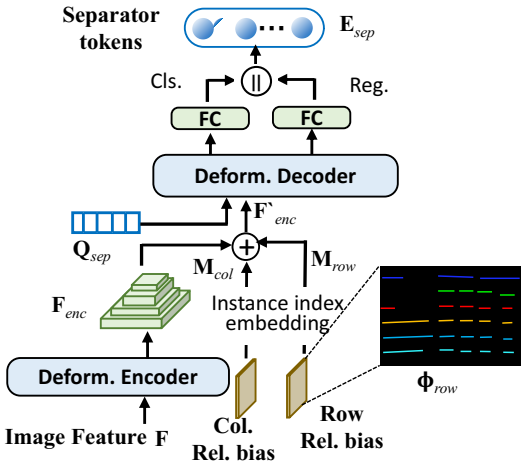

Figure 3: Separator Perceiver. Best viewed in color.

T sample points $\{\mathcal{S}_i(x_i, y_i)\}_{i=1}^{T}$, where curve length between each of adjacent are equal. Hence, our method tries to regress these sample points. Once the points determined, the control points of whole curve can be easily acquired by standard "least squares fitting" algorithm (Weisstein 2002).

Separator Perceiver. Now, we elaborate on how to generate separator proposals with our newly designed Separator Perceiver (SP). Specifically, as illustrated in Fig. 3, considering the scale variety of separators, we build SP upon Deformable DETR (Zhu et al. 2020), where Deformable encoder takes table image feature $\mathbf{F}$ as input. Here, we adopt the canonical ResNet-50 (He et al. 2016) as the image feature extractor. Then, the multi-scale encoder feature $\mathbf{F}_{enc} = \{D_1, D_2, D_3, D_4\}$ is output, which respectively have strides of 8, 16, 32, 64 pixels with respect to the raw table image in H height and W width. To facilitate the learning of separator, the relation component is converted to the column and row relation biases ($\mathbf{M}_{\sim} \in \{\mathbf{M}_{row}, \mathbf{M}_{col}\}$) applied on the encoder feature $\mathbf{F}_{enc}$. Taking the row relation bias for example, the row relation mask $\Phi_{row} \in \mathbb{R}^{W \times H}$ is firstly generated according to $\mathbf{R}_{row}$. To be specific, if two elements belong to the same row relations, the pixels inside the element bounding boxes will be assigned with the same relation instance integral index $r, r \in 1, 2, ..., M$ (denoted by the same color in Fig. 3, while regions outside element boxes are assigned 0 value. After down-sampled to the size of each scale feature, we embed these multi-scale maps by embedding method (Mikolov et al. 2013) to obtain $\mathbf{M}_{row}$. Then, both $\mathbf{M}_{row}$ and $\mathbf{M}_{col}$ are added to each scale of encoder feature as $\mathbf{F}'_{enc}$, which is sent to the deformable decoder module subsequently. For the decoder, we adopt Q learnable separator queries $\mathbf{Q}_{sep} \in \mathbb{R}^{Q \times d_q}$ to predict a set of sample points $\{\mathcal{S}_i(x_i, y_i)\}_{i=1}^{T}$ representing each separator. Here, the feature output by corresponding FC (Fully-Connected) layers of sample points regression and classification are concatenated as separator tokens $\mathbf{E}_{sep} \in \mathbb{R}^{Q \times (2 \cdot T + 3)}$, where $2 \cdot T$ channels correspond to the x/y coordinates of T sample points while 3 channels correspond to classes of explicit separators, implicit separators and background.

Multiple Components Correlation

During this phase, the aim is to correlate the candidate components $\mathbf{E}_{ele}$ and $\mathbf{E}_{rel}$ to the main clue $\mathbf{E}_{sep}$. To implement above purpose, we build our Components Correlator (CC) upon the canonical transformer (Vaswani et al. 2017) consisting of FeedForward Network (FFN) and Multi-head Cross Attention (MHCA) interleaved with Layer Normalization and residual connection. Different from the vanilla one, our CC aggregates the both candidate components in a decoupled way to avoid the interference between the information of large difference they carry. More concretely, the $\mathbf{E}_{sep}$ play as roles of queries ($\mathbf{Q}_1$ and $\mathbf{Q}_2$) sent to the both streams, where each of them treats either $\mathbf{E}_{ele}$ or $\mathbf{E}_{rel}$ as the key and value for the MHCA. The block is stacked by N times. Finally, similar to the separator prediction, we also append the 3-dimension classification and $2T$-dimension regression layers, where the respective FC features are concatenated as the enhanced separator tokens $\widetilde{\mathbf{E}}_{sep}$. Note, all the queries and keys and values are added with position encoding (PE) (Vaswani et al. 2017). Through this way, the separator proposals represented by separator tokens could be refined and aggregated with appropriate exclusive components, which is essential to the final structure prediction.

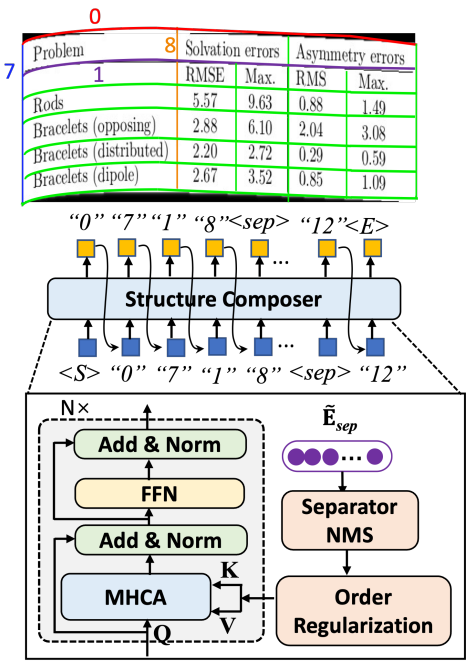

Figure 4: Structure Composer. Best viewed in color.

Table Structure Composing

At the final stage of "deliberation" process, the refined separator proposals are treated as the ingredients to compose the final table cell in logical order. Motivated by Pix2Seq (Chen et al. 2021), we propose a novel generative Structure Composer (SC) to define above process as a sequence-to-sequence generation problem, based on a intuition that if a model knows about where and what the separators are, we just need to teach it how to compose them as cells. As shown in Fig. 4, instead of directly predicting the cell coordinates widely-adopted in most methods, our SC predicts a sequence of selected separator indexes where each four corresponding separators construct a cell's boundary.

Formally, the output sequence $\mathcal{G}$ is represented as $\{\langle S\rangle, C_0, \langle sep\rangle, C_1, ..., C_g, \langle E\rangle\}$, where $C_{\sim} = [\alpha_{top}, \alpha_{left}, \alpha_{bottom}, \alpha_{right}]$ are the anticlockwise arranged indexes of separators wrapping a cell. $\langle S\rangle$, $\langle sep\rangle$ and $\langle E\rangle$ are start, separation and end tokens respectively. Given a start token "$\langle S\rangle$" as query, the SC recursively predicts index sequence $\mathcal{G}$ until the "$\langle E\rangle$" is output. The basic block of SC is also built on the vanilla transformer block (Vaswani et al. 2017), where enhanced separator tokens $\widetilde{\mathbf{E}}_{sep}$ are regarded as keys and values with PE added.

To control the input sequence length to a reasonable ac-

count, before sent to the transformer blocks, $\widetilde{\mathbf{E}}_{sep}$ and corresponding separator proposals are firstly processed by the "Separator NMS". As the original NMS (Ren et al. 2015) is designed for the object detection task, to adapt it to the TSR task, we repurpose it into our "Separator NMS" by replacing the IoU metric with our proposed Separator Distance:

$$\mathcal{D}_{sep} = \frac{1}{T}\sum_{t\in T} ||\mathcal{S}_t^{(i)}(x_t, y_t) - \mathcal{S}_t^{(j)}(x_t, y_t)||_2, \quad (1)$$

where $\mathcal{S}_t^{(i)}(x_t, y_t)$ is the t-th sample points from the i-th separator (corresponds to the i-th token of $\widetilde{\mathbf{E}}_{sep}$). The condition for removing separator is set as $\mathcal{D}_{sep} < \sigma$, where σ is set to 5 in default. Considering the selected separators are unordered, which could cause training collapse problem, we sent the selected separators to the "Order Regularization" submodules. They are firstly grouped into horizontal ($\mathcal{H}_h$) and vertical ($\mathcal{H}_v$) types according to the slope of the first (start) sample point and last (end) one connected line. The output of "Order Regularization" is defined as:

$$\mathcal{H} = \{Sort_y(\mathcal{H}_h), Sort_x(\mathcal{H}_v)\}, \quad (2)$$

where "$Sort_x$" and "$Sort_y$" denotes sort operations along x and y axes respectively.

Training Strategy

Loss function. As illustrated in Fig. 2, our proposed GrabTab is trained in an end-to-end way by multi-task loss $\mathcal{L} = \lambda_1\mathcal{L}_{sep_1} + \lambda_2\mathcal{L}_{sep_2} + \lambda_3\mathcal{L}_{struct}$, where balance weight of each loss $\lambda_\sim$ is set to 1 equally. Among, the separator loss $\mathcal{L}_{sep\sim}$ is the modified Hungarian matching loss (Zhu et al. 2020) adapting to the TSR task:

$$\mathcal{L}_{sep_\sim}(O, G) = \min_{\gamma} \underbrace{\left(\sum_{i=1}^{M} \mathcal{L}_{cls}(c_{\gamma_i}, \hat{c}_i) + \mathcal{L}_{reg}(s_{\gamma_i}, \hat{s}_i)\right)}_{\mathcal{L}(O,G|\gamma)}. \quad (3)$$

Similar to (Zhu et al. 2020), during training, the model firstly assigns each output $O = \{s_j, c_j\}_{j=1}^{M}$ to exactly one annotated separator in $G = \{\hat{s}_k, \hat{c}_k\}_{k=1}^{2}$ or background $\emptyset$, where c_k is the k-th class (explicit or implicit separator). Note that O and G are both denoted by sample points. $\gamma_i \in \{\emptyset, 1, 2\}$ is the assignment of model output i to ground truth (GT) γ_i, while $\mathcal{L}_{reg}$ is implemented by Eqn. (1). For the $\mathcal{L}_{struct}$, we adopt the standard cross-entropy loss which is similar with other generative models (Chen et al. 2021).

Separator assignment. Nevertheless, the above one-to-one assignment in vanilla Hungarian matching algorithm would cause the severe separator missing problem, due to the slim shape of separator. To attack it, inspired by method (Ouyang-Zhang et al. 2022), we further modify the assignment to the one-to-many strategy, *i.e.*, one GT is assigned with a group rather one separator. Based on the one-to-one assigned separator $\{s_{\gamma_i}, c_{\gamma_i}\}$, we further find its neighboring ones on the condition that $\mathcal{D}_{sep} < 5$. Here, $\mathcal{D}_{sep}$ is the Separator Distance (defined in Eqn. (1)) between $\{s_{\gamma_i}, c_{\gamma_i}\}$ and its neighboring separators. Afterwards, the grouped separators are also assigned to the GT annotation γ_i.

Experiments

Datasets and Evaluation Protocol

Datasets. We evaluate our method on the following benchmark datasets under both complex and regularized table scenarios. ICDAR-2013 (Göbel et al. 2013), ICDAR-2019 (Gao et al. 2019), WTW (Long et al. 2021), UNLV (Shahab et al. 2010), SciTSR (Chi et al. 2019), SciTSR-COMP (Chi et al. 2019) and SciTSR-COMP-A (Liu et al. 2022) are evaluated under protocol of physical structure recognition, while TableBank (Li et al. 2020) and PubTabNet (Zhong, ShafieiBavani, and Jimeno Yepes 2020) are adopted to evaluate the logical structure recognition performance. Furthermore, WTW (Long et al. 2021), SciTSR-COMP (Chi et al. 2019) and SciTSR-COMP-A (Liu et al. 2022) are employed as complex TSR datasets with more challenging distractors involved, while the rest are the regularized ones.

Evaluation protocol. For a fair comparison, we inherit the widely-adopted protocols from prevalent methods. Among, precision, recall and F1-score are utilized to evaluate the performance of recognizing table physical structure. And the performance of table logical structure recognition is evaluated by the Tree-Edit-Distance-based Similarity (TEDS)-Struct (Zhong, ShafieiBavani, and Jimeno Yepes 2020) and BLEU score (Papineni et al. 2002) protocols.

Implementation Details

We build the framework using Pytorch (Paszke et al. 2019) and conduct all experiments on a workstation with 8 Nvidia Tesla V100 GPUs. All the component tokens are projected into 256-dimensional vertors. All the transformer block numbers in SP, CC and SC are set to 6, where the dimensions of hiddens and FFN are 256 and 2,048 respectively. The number of queries in SP is empirically set to 1,000. The framework is optimized by AdamW (Loshchilov and Hutter 2017) with a batch size of 16. We adopt the learning rate $1\mathrm{e}{-5}$ for both Seperator Perceiver and Components Correlator, and $1\mathrm{e}{-4}$ for Structure Composer. The number of sample points T for representing separator is set to 15. During the training phase, the table images within a same batch are randomly resized ranging from 480 to 800 while the size is fixed to 1,100 for test. For all experiments, the network is pre-trained on SciTSR for 10 epochs, and then fine-tuned on different benchmarks for 50 epochs.

Comparison with State-of-the-arts

Results of complex table structure recognition. Tab. 1 gives comparison results on several benchmark datasets. Among, the first three columns indicate the performance on recognizing complex tables containing more severe distractors. Compared with existing methods, the F1-score of our GrabTab can beat the second best method, NCGM (Liu et al. 2022), by 4.5% on SciTSR-COMP-A dataset, while the apparent performance improvement is also witnessed on WTW and SciTSR-COMP datasets. This phenomenon further confirms that simply focusing on single component extraction is not the optimal solution. By equipped with deliberation mechanism, our GrabTab finds the silver linings behind the

	WTW				SciTSR-COMP				SciTSR-COMP-A				TableBank		PubTabNet	
Method	Train Set	P	R	F1	Train Set	P	R	F1	Train Set	P	R	F1	Train Set	BLEU	Train Set	TEDS
C-CTRNet	WTW	93.3	91.5	92.4	-	-	-	-	-	-	-	-	-	-	-	-
FLAG-Net	WTW	91.6	89.5	90.5	SciTSR	98.4	98.6	98.5	Sci. + Sci.-C-A	82.5	83.0	82.7	SciTSR	93.9	SciTSR	95.1
TSRFormer	WTW	94.5	94.0	94.3	SciTSR	**99.1**	98.6	98.9	-	-	-	-	-	-	PubTabNet	97.5
RobusTabNet	WTW	-	-	-	SciTSR	99.0	98.4	98.7	-	-	-	-	-	-	PubTabNet	97.0
TableFormer	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PubTabNet	96.8
LGPMA	WTW	91.3	88.9	90.1	SciTSR	97.3	98.7	98.0	Sci. + Sci.-C-A	81.8	82.2	82.0	SciTSR	93.5	PubTabNet	96.7
NCGM	WTW	93.7	94.6	94.1	SciTSR	98.7	98.9	98.8	Sci. + Sci.-C-A	88.4	90.7	89.5	SciTSR	94.6	SciTSR	95.4
GrabTab	WTW	**95.3**	**95.0**	**95.1**	SciTSR	98.9	**99.4**	**99.1**	Sci. + Sci.-C-A	**94.3**	**93.8**	**94.0**	SciTSR	**95.0**	PubTabNet	**97.9**

Table 1: Comparison results with SOTAs (C-CTRNet (Long et al. 2021), FLAG-Net (Liu et al. 2021), TSRFormer (w/ DQ-DETR) (Wang et al. 2023), TableFormer (Nassar et al. 2022), RobusTabNet (Ma et al. 2023), NCGM (Liu et al. 2022) and LGPMA (Qiao et al. 2021)) on WTW, SciTSR-COMP, SciTSR-COMP-A, TableBank and PubTabNet datasets. "P", "R" and "F1" stand for "Precision", "Recall" and "F1-score" respectively. "C-CTRNet" and "Sci." are short for "Cycle-CenterNet" and SciTSR.

problem of multiple components leverage, with decent results produced.

Results of regularized table structure recognition. In the last two columns of Tab. 1, the performance on regularized table cases is also given. As they are evaluated under logical structure recognition protocol, we convert the output physical structure format to the HTML, which strictly follows the operation in NCGM. From the table, one can observe that, on both datasets, our GrabTab can also achieve the consistent improvement than the state-of-the-arts.

Ablation Study

In this subsection, we investigate the effects of various factors in GrabTab by juxtaposing analytic experiments on SciTSR-COMP-A dataset, which is the most challenging complex dataset. If we remove all the tailored designs in our GrabTab, it would degenerate to the "Deform.-DETR" regarding separators as prediction targets. By simply adding relation biases ($\mathbf{M}_{\sim} \in \{\mathbf{M}_{row}, \mathbf{M}_{col}\}$ in "separator proposals generation", we surprisingly observe the 4.1% F1-score improvement brought by "GrabT.$_{\text{w/o CC/MA/SC}}$". Moreover, if the Components Correlator (CC) submodule is appended, the performance could be further boosted by "GrabT.$_{\text{w/o MA/SC}}$", which is a persuasive evidence to demonstrate the effectiveness of CC.

Method	RB	CC	OA	MA	RP	SC	P	R	F1
Deform.-DETR	✗	✗	✓	✗	✓	✗	84.8	85.1	84.9
GrabT.$_{\text{w/o CC/MA/SC}}$	✓	✗	✓	✗	✓	✗	88.7	89.4	89.0
GrabT.$_{\text{w/o MA/SC}}$	✓	✓	✓	✗	✓	✗	90.2	90.4	90.3
GrabT.$_{\text{w/o SC}}$	✓	✓	✗	✓	✓	✗	91.5	91.8	91.6
GrabTab-S	✓	✓	✗	✓	✗	✓	91.9	92.4	92.1
GrabTab	✓	✓	✗	✓	✗	✓	**94.3**	**93.8**	**94.0**

Table 2: Ablation studies of GrabTab on SciTSR-COMP-A dataset. Legend: "RB": Row/Column Relation Bias, "CC": Components Correlator, "OA": One-to-One Assignment, "MA": One-to-Many Assignment, "RP": Rule-basd Post-processing, "SC": Structure Composer. "w/o." is short for "without" and "-S" represents straight-line separators.

In addition to relation biases (RB) and CC, separator assignment way is another factor of importance. By adopting original one-to-one assignment (OA) trick, the F1 performance witnesses 1.3% drop compared with the version ("GrabT.$_{\text{w/o SC}}$") equipped with our modified one-to-many assignment (MA). We attribute the performance drop to the "separator missing" problem, *i.e.*, most separators are not perceived under original one-to-one assignment.

As elaborated above, most existing methods depend on complicated post-processing based on heuristic, which is not versatile for various complex tables. Comparatively, our Structure Composer (SC) is able to learn how to pick up desired separators and compose them into the final structure according to the specific cases. As indicated by "'GrabTab", which is the full version of our method, by equipping it with SC, the F1-score can be increased to 94.0%, which surpasses the rule-based version ("GrabT.$_{\text{w/o SC}}$") by a large margin. To investigate the effect of separator quality, we further modify the prediction targets in "GrabTab-S", where target separators are represented in straight-line format instead. Consequently, we find the modification is detrimental to the performance, which can be attributable to the inconsistence between target curved representation and predicted straight lines, especially highlighted for the distorted tables.

Further Analysis on Deliberation

What components matter during deliberation? To investigate the behaviors of different components during deliberation, for the sample from SciTSR-COMP-A dataset, we in Fig. 5 visualize the attention heat-maps from last block of Components Correlator module to reflect the correlations between separator lines and candidate components, *i.e.*, elements (orange boxes) and their relations of row (blue connections)/column (purple connections). For clarity, we pick up one explicit (red solid curve) and implicit (blue dashed curve) separator line as examples, and directly draw the correlations on the raw table images. The darker color indicates stronger correlation.

In the left part of Fig. 5, explicit lines, e.g., "L4", demonstrate a broad attention span, attending to nearly all elements with a smooth distribution of attention weights. On

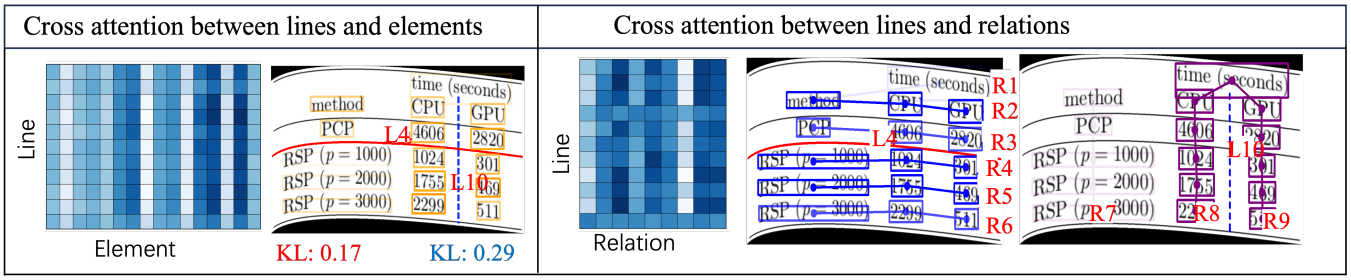

Figure 5: Visualizations of the attention heat-maps from last block of Components Correlator. The $L_{\sim}$ and $R_{\sim}$ correspond to lines and relations. Table elements or relation connections with darker colors indicate stronger correlations with exemplar lines. Best viewed in color and zoomed in.

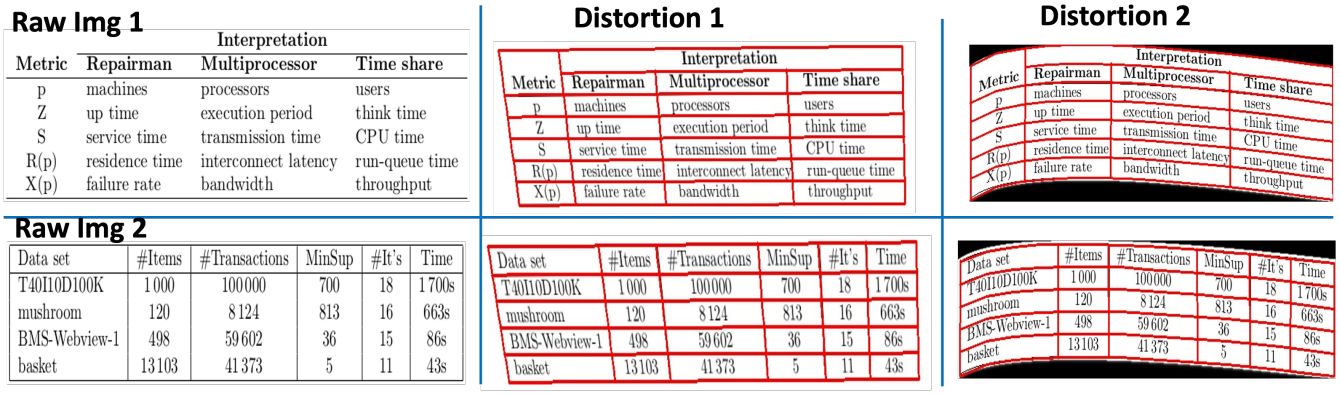

Figure 6: Qualitative results on SciTSR-COMP-A.

the other hand, implicit lines e.g., "L10", tend to establish relationships with nearby spanning elements, indicating that adjacent elements contribute more to recovering invisible implicit lines. We quantitatively compare the attention patterns of explicit and implicit lines using Kullback-Leibler divergence (Joyce 2011), which confirms that implicit lines have a higher average divergence (0.29 vs. 0.17), indicating stronger correlations with elements. This correlation leads to consistent performance improvement.

On the other hand, in order to show the cooperation patterns between lines and relations, the cross attentions between lines/row-relations and lines/column-relations are separately visualized in the right part of Fig. 5. Obviously, the explicit lines prefer those relations carrying the same span information ("L4" related "R2"∼"R9"), while the implicit lines pay more attention to the adjacent local relations ("L10" related from "R8" to "R9"), which vividly illustrates the importance of relational features to the reconstruction of invisible separators. Simultaneously, the effectiveness of the CC module for "grabbing" informative clues between lines and relations is also verified.

Qualitative results. We in Fig. 6 visualize several recognition results from the complex TSR datasets, SciTSR-COMP-A (Liu et al. 2022), which further confirm the superior performance of our GrabTab. Among, the tables in SciTSR-COMP-A are distracted by not only inherent table structures, but also two kinds of synthesized distortions, *i.e.*, affine transformation and curverd distortion. From the results shown in Fig. 6, one can observe that our method can precisely predict the distorted table cells, even with severe content misalignment.

Conclusion and Limitations

In this work, we introduce GrabTab, a method that utilizes a deliberation mechanism to handle complex tables with multiple assembled components, eliminating the need for complicated post-processing. This exclusive mechanism ensures the robustness and versatility of our method. We conduct extensive experiments on various table datasets to validate and analyze its performance. The experimental results demonstrate that our model outperforms previous approaches, particularly in complex table scenarios, confirming the effectiveness of our method. As for the limitations of our work, in the current version, we simply extract the table elements and and their relationships in off-line manner. Alternatively, a more potential way is to repurpose the candidate components extraction into an on-line one, with the model weight updated together with the subsequent deliberator. Besides, the increase on the computational complexity is another issue to be solved we leave in our future work.

References

Anderson, J. R. 2005. *Cognitive psychology and its implications*. Macmillan.

Chen, T.; Saxena, S.; Li, L.; Fleet, D. J.; and Hinton, G. 2021. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*.

Chi, Z.; Huang, H.; Xu, H.-D.; Yu, H.; Yin, W.; and Mao, X.-L. 2019. Complicated Table Structure Recognition. *arXiv preprint arXiv:1908.04729*.

Clancey, W. J. 2002. Simulating activities: Relating motives, deliberation, and attentive coordination. *Cognitive Systems Research*, 3(3): 471–499.

Feng, H.; Liu, Q.; Liu, H.; Zhou, W.; Li, H.; and Huang, C. 2023a. DocPedia: Unleashing the Power of Large Multimodal Model in the Frequency Domain for Versatile Document Understanding. *arXiv preprint arXiv:2311.11810*.

Feng, H.; Wang, Z.; Tang, J.; Lu, J.; Zhou, W.; Li, H.; and Huang, C. 2023b. Unidoc: A universal large multimodal model for simultaneous text detection, recognition, spotting and understanding. *arXiv preprint arXiv:2308.11592*.

Gao, L.; Huang, Y.; Déjean, H.; Meunier, J.-L.; Yan, Q.; Fang, Y.; Kleber, F.; and Lang, E. 2019. ICDAR 2019 competition on table detection and recognition (cTDaR). In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1510–1515. IEEE.

Göbel, M.; Hassan, T.; Oro, E.; and Orsi, G. 2013. ICDAR 2013 table competition. In *2013 12th International Conference on Document Analysis and Recognition*, 1449–1453. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Jauhar, S. K.; Turney, P.; and Hovy, E. 2016. Tables as semi-structured knowledge for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 474–483.

Joyce, J. M. 2011. Kullback-leibler divergence. In *International encyclopedia of statistical science*, 720–722. Springer.

Khan, S. A.; Khalid, S. M. D.; Shahzad, M. A.; and Shafait, F. 2019. Table structure extraction with bi-directional gated recurrent unit networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1366–1371. IEEE.

Li, J.; Monroe, W.; Ritter, A.; Jurafsky, D.; Galley, M.; and Gao, J. 2016. Deep Reinforcement Learning for Dialogue Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1192–1202.

Li, M.; Cui, L.; Huang, S.; Wei, F.; Zhou, M.; and Li, Z. 2020. Tablebank: Table benchmark for image-based table detection and recognition. In *Proceedings of The 12th Language Resources and Evaluation Conference*, 1918–1925.

Liu, H.; Li, X.; Liu, B.; Jiang, D.; Liu, Y.; and Ren, B. 2022. Neural collaborative graph machines for table structure recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4533–4542.

Liu, H.; Li, X.; Liu, B.; Jiang, D.; Liu, Y.; Ren, B.; and Ji, R. 2021. Show, Read and Reason: Table Structure Recognition with Flexible Context Aggregator. In *Proceedings of the 29th ACM International Conference on Multimedia*, 1084–1092.

Liu, Y.; Chen, H.; Shen, C.; He, T.; Jin, L.; and Wang, L. 2020. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9809–9818.

Long, R.; Wang, W.; Xue, N.; Gao, F.; Yang, Z.; Wang, Y.; and Xia, G.-S. 2021. Parsing Table Structures in the Wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 944–952.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ma, C.; Lin, W.; Sun, L.; and Huo, Q. 2023. Robust table detection and structure recognition from heterogeneous document images. *Pattern Recognition*, 133: 109006.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nassar, A.; Livathinos, N.; Lysak, M.; and Staar, P. 2022. Tableformer: Table structure understanding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4614–4623.

Olshausen, B. A.; Anderson, C. H.; and Van Essen, D. C. 1993. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13(11): 4700–4719.

Ouyang-Zhang, J.; Cho, J. H.; Zhou, X.; and Krähenbühl, P. 2022. NMS Strikes Back. *arXiv preprint arXiv:2212.06137*.

Paliwal, S. S.; Vishwanath, D.; Rahul, R.; Sharma, M.; and Vig, L. 2019. Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 128–133. IEEE.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Qasim, S. R.; Mahmood, H.; and Shafait, F. 2019. Rethinking table recognition using graph neural networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 142–147. IEEE.

Qiao, L.; Li, Z.; Cheng, Z.; Zhang, P.; Pu, S.; Niu, Y.; Ren, W.; Tan, W.; and Wu, F. 2021. LGPMA: Complicated Table

Structure Recognition with Local and Global Pyramid Mask Alignment. *arXiv preprint arXiv:2105.06224*.

Raja, S.; Mondal, A.; and Jawahar, C. 2020. Table Structure Recognition using Top-Down and Bottom-Up Cues. In *European Conference on Computer Vision*, 70–86. Springer.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Schreiber, S.; Agne, S.; Wolf, I.; Dengel, A.; and Ahmed, S. 2017. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, 1162–1167. IEEE.

Shahab, A.; Shafait, F.; Kieninger, T.; and Dengel, A. 2010. An open approach towards the benchmarking of table structure recognition systems. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, 113–120.

Tensmeyer, C.; Morariu, V. I.; Price, B.; Cohen, S.; and Martinez, T. 2019. Deep splitting and merging for table structure decomposition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 114–121. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is All you Need. *Advances in Neural Information Processing Systems*, 30: 5998–6008.

Wang, J.; Lin, W.; Ma, C.; Li, M.; Sun, Z.; Sun, L.; and Huo, Q. 2023. Robust Table Structure Recognition with Dynamic Queries Enhanced Detection Transformer. *arXiv preprint arXiv:2303.11615*.

Weisstein, E. W. 2002. Least squares fitting. *https://mathworld. wolfram. com/*.

Zheng, X.; Burdick, D.; Popa, L.; Zhong, X.; and Wang, N. X. R. 2021. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 697–706.

Zhong, X.; ShafieiBavani, E.; and Jimeno Yepes, A. 2020. Image-based table recognition: data, model, and evaluation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, 564–580. Springer.

Zhong, X.; ShafieiBavani, E.; and Yepes, A. J. 2019. Image-based table recognition: data, model, and evaluation. *arXiv preprint arXiv:1911.10683*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.