A Hybrid Approach to Document Layout Analysis for Heterogeneous Document Images

Zhuoyao Zhong[1(✉)], Jiawei Wang[1,2], Haiqing Sun[1,3], Kai Hu[1,2], Erhan Zhang[1,3], Lei Sun[1], and Qiang Huo[1]

[1] Microsoft Research Asia, Beijing, China
zhuoyao.zhong@gmail.com, {wangjiawei,hk970213}@mail.ustc.edu.cn
[2] Department of EEIS, University of Science and Technology of China, Hefei, China
[3] School of Software and Microelectronics, Peking University, Beijing, China
{sunhq5,zhangeh-ss}@stu.pku.edu.cn

Abstract. We present a new hybrid document layout analysis approach to simultaneously detecting graphical page objects, group text-lines into text regions according to reading order, and recognize the logical roles of text regions from heterogeneous document images. For graphical page object detection, we leverage a state-of-the-art Transformer-based object detection model, namely DINO, as a new graphical page object detector to detect tables, figures, and (displayed) formulas in a top-down manner. Furthermore, we introduce a new bottom-up text region detection model to group text-lines located outside graphical page objects into text regions according to reading order and recognize the logical role of each text region by using both visual and textual features. Experimental results show that our bottom-up text region detection model achieves higher localization and logical role classification accuracy than previous top-down methods. Moreover, in addition to the locations of text regions, our approach can also output the reading order of text-lines in each text region directly. The state-of-the-art results obtained on DocLayNet and PubLayNet demonstrate the effectiveness of our approach.

Keywords: Document layout analysis · Graphical page object detection · Text region detection · Reading order prediction

1 Introduction

Document layout analysis is the process of recovering document physical and/or logical structures from document images, including physical layout analysis and logical layout analysis [10]. Given input document images, physical layout analysis aims at identifying physical homogeneous regions of interest (also called

J. Wang, H. Sun, K. Hu and E. Zhang—This work was done when Jiawei Wang, Haiqing Sun, Kai Hu and Erhan Zhang were interns in MMI Group, Microsoft Research Asia, Beijing, China.

G. A. Fink et al. (Eds.): ICDAR 2023, LNCS 14191, pp. 189–206, 2023.
https://doi.org/10.1007/978-3-031-41734-4_12

page objects), such as graphical page objects like tables, figures and formulas, and different types of text regions. Logical layout analysis aims at assigning a logical role to the identified regions (e.g., title, section heading, header, footer, paragraph) and determining their logical relationships (e.g., reading order relationships and key-value pair relationships). Document layout analysis plays an important role in document understanding, which can enable a wide range of applications, such as document digitization, conversion, archiving, and retrieval. However, owing to the diverse contents and complex layouts of documents, large variability in region scales and aspect ratios, and similar visual textures between different types of text regions, document layout analysis is still a challenging problem.

In recent years, many deep learning based document layout analysis approaches have emerged [17,22,36,46,47,51,53] and substantially outperformed traditional rule based or handcrafted feature based methods in terms of both accuracy and capability [3]. These methods usually borrow existing object segmentation and detection models, like FCN [29], Faster R-CNN [37], Mask R-CNN [15], Cascade R-CNN [5], SOLOv2 [44], Deformable DETR [54], to detect target page objects from document images. Although they have achieved superior results on graphical page object detection, their performance on text region detection is still unsatisfactory. First, these methods cannot detect small-scale text regions that only span one or two text-lines (e.g., header, footer, and section headings) with high localization accuracy. For example, the detection accuracy of DINO, which is a state-of-the-art Transformer-based detection model [49], for these small-scale text regions drops more than 30% when the Intersection-over-Union (IoU) threshold is increased from 0.5 to 0.75 on DocLayNet [36] based on our experimental results. Second, when two different types of text regions have similar visual textures, e.g., paragraphs and list items, paragraphs and section headings, and section headings and titles, these methods cannot distinguish them robustly. Moreover, these methods only extract the boundaries or masks of text regions and cannot output the reading order of text-lines in text regions, which makes the outputs of these methods hard to be consumed by some important downstream applications such as translation and information extraction.

To address these issues, we present a new hybrid document layout analysis approach to simultaneously detecting graphical page objects, group text-lines into text regions according to reading order, and recognize the logical roles of text regions from heterogeneous document images. For graphical page object detection, we propose to leverage the DINO [49] as a new graphical page object detector to detect tables, figures, and (displayed) formulas in a top-down manner. Furthermore, we introduce a new bottom-up text region detection model to group text-lines located outside graphical page objects into text regions according to reading order and recognize the logical role of each text region by using both visual and textual features. The DINO-based graphical page object detection model and the new bottom-up text region detection model share the same CNN backbone network so that the whole network can be trained in an end-to-end manner. Experimental results demonstrate that this new bottom-up text

region detection model can achieve higher localization accuracy for small-scale text regions and better logical role classification accuracy than previous top-down text region detection approaches. Moreover, in addition to the locations of text regions, our approach can also output the reading order of text-lines in each text region directly. State-of-the-art results obtained on two large-scale document layout analysis datasets (i.e., DocLayNet [36] and PubLayNet [53]) demonstrate the effectiveness and superiority of our approach. Especially on DocLayNet, our approach outperforms the previous best-performing model by 4.2% in terms of mean Average Precision (mAP). Although PubLayNet has been well-tuned by many previous techniques, our approach still achieves the highest mAP of 96.5% on this dataset.

2 Related Work

A comprehensive survey of traditional document layout analysis methods has been given in [3]. In this section, we focus on reviewing recent deep learning based approaches that are closely related to this work. These approaches can be roughly divided into three categories: object detection based methods, semantic segmentation based methods, and graph-based methods.

Object Detection Based Methods. These methods leverage state-of-the-art top-down object detection or instance segmentation frameworks to address the document layout analysis problem. Yi et al. [48] and Oliveira et al. [35] first adapted R-CNN [12] to locate and recognize page objects of interest from document images, while their performance was limited by the traditional region proposal generation strategies. Later, more advanced object detectors, like Fast R-CNN [11], Faster R-CNN [37], Mask R-CNN [15], Cascade R-CNN [5], SOLOv2 [44], YOLOv5 [18], Deformable DETR [54], were explored by Vo et al. [42], Zhong et al. [53], Saha et al. [38], Li et al. [20], Biswas et al. [4], Pfitzmann et al. [36], and Yang et al. [46], respectively. Meanwhile, some effective techniques were also proposed to further improve the performance of these detectors. For instance, Zhang et al. [51] proposed a multi-modal Faster/Mask R-CNN model to detect page objects, in which visual feature maps extracted by CNN and two 2D text embedding maps with sentence and character embeddings were fused together to construct multi-modal feature maps and a graph neural network (GNN) based relation module was introduced to model the interactions between page object candidates. Bi et al. [2] also proposed to leverage GNN to model the interactions between page object candidates to improve page object detection accuracy. Naik et al. [34] incorporated the scale-aware, spatial-aware, and task-aware attention mechanisms proposed in DynamicHead [8] into the CNN backbone network to improve the accuracy of Faster R-CNN and Sparse R-CNN [41] for page object detection. Shi et al. [40] proposed a new lateral feature enhancement backbone network and Yang et al. [46] leveraged Swin Transformer [28] as a stronger backbone network to push the performance of Mask R-CNN and Deformable DETR on page object detection tasks, respectively. Recently, Gu et al. [13], Li et al.

[20] and Huang et al. [17] improved the performance of Faster R-CNN, Mask R-CNN, and Cascade R-CNN based page object detectors by pre-training the vision backbone networks on large-scale document images with self-supervised learning algorithms. Although these methods have achieved state-of-the-art performance on several benchmark datasets, they still struggle with small-scale text region detection and cannot output the reading order of text-lines in text regions directly.

Semantic Segmentation Based Methods. These methods (e.g., [14,23,24,39,47]) usually use existing semantic segmentation frameworks like FCN [29] to predict a pixel-level segmentation mask first, and then merge pixels into different types of page objects. Yang et al. [47] proposed a multi-modal FCN for page object segmentation, where visual feature maps and 2D text embedding maps with sentence embeddings were combined to improve pixel-wise classification accuracy. He et al. [14] proposed a multi-scale multi-task FCN to simultaneously predict a region segmentation mask and a contour segmentation mask. After being refined by a conditional random field (CRF) model, these two segmentation masks are then input to a post-processing module to get final prediction results. Li et al. [23] incorporated label pyramids and deep watershed transformation into the vanilla FCN to avoid merging nearby page objects together. The performance of existing semantic segmentation based methods is still inferior to the other two types of methods on recent document layout analysis benchmarks.

Graph-Based Methods. These methods (e.g., [21,22,32,43]) model each document page as a graph whose nodes represent primitive page objects (e.g., words, text-lines, connected components) and edges represent relationships between neighboring primitive page objects, and then formulate document layout analysis as a graph labeling problem. Li et al. [21] used image processing techniques to generate line regions first, and then applied two CRF models to classify them into different types and predict whether each pair of line regions belong to the same instance based on visual features extracted by CNNs, respectively. Based on these prediction results, line regions belonging to the same class and the same instance were merged to get page objects. In their subsequent work [22], they used connected components to replace line regions as nodes and adopted a graph attention network (GAT) to enhance the visual features of both nodes and edges. Luo et al. [32] focused on the logical role classification task and proposed to leverage syntactic, semantic, density, and appearance features with multi-aspect graph convolutional networks (GCNs) to recognize the logical role of each page object. Recently, Wang et al. [43] focused on the paragraph identification task and proposed a GCN-based approach to grouping text-lines into paragraphs. Liu et al. [27], Long et al. [30] and Xue et al. [45] further proposed a unified framework for text detection and paragraph (text-block) identification.

Unlike these works, our unified layout analysis approach can detect page objects, predict the reading order of text-lines in text regions and recognize the logical roles of text regions from document images simultaneously.

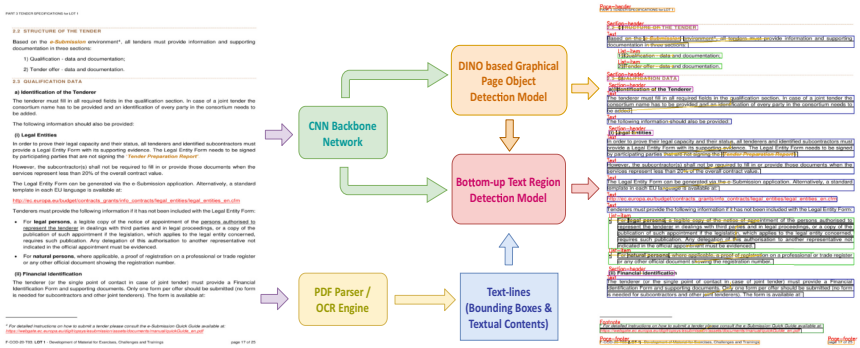

Fig. 1. The overall architecture of our hybrid document layout analysis approach.

3 Methodology

Our approach is composed of three key components: 1) A shared CNN backbone network to extract multi-scale feature maps from input document images; 2) A DINO based graphical page object detection model to detect tables, figures, and displayed formulas; 3) A bottom-up text region detection model to group text-lines located outside graphical page objects into text regions according to reading order and recognize the logical role of each text region. These three components are jointly trained in an end-to-end manner. The overall architecture of our approach is depicted in Fig. 1. The details of these components are described in the following subsections.

3.1 Shared CNN Backbone Network

Given an input document image $I \in \mathbb{R}^{H \times W \times 3}$, we adopt a ResNet-50 network [16] as the backbone network to generate multi-scale feature maps $\{C_3, C_4, C_5\}$, which represent the output feature maps of the last residual block in Conv3, Conv4, and Conv5, respectively. C_6 is obtained via a 3×3 convolutional layout with stride 2 on C_5. The resolutions of $\{C_3, C_4, C_5, C_6\}$ are $1/8, 1/16, 1/32, 1/64$ of the original document image. Then, a 1×1 convolutional layer is performed on each feature map for channel reduction. After that, all feature maps have 256 channels, which are input to the following DINO based graphical page object detection model and bottom-up text region detection model.

3.2 DINO Based Graphical Page Object Detection Model

Recently, Transformer-based object detection methods such as DETR [6], Deformable DETR [54], DAB-DETR [26], DN-DETR [19] and DINO [49] have become popular as they can achieve better performance than previous

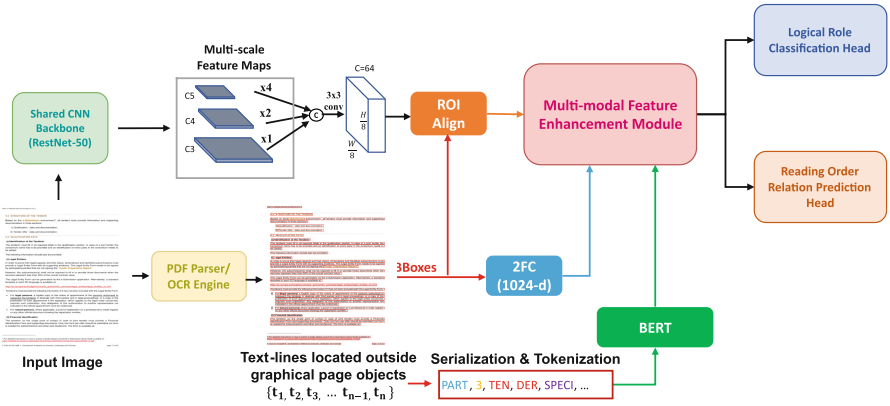

Fig. 2. A schematic view of the proposed bottom-up text region detection model.

Faster/Mask R-CNN based models without relying on manually designed components like non-maximum suppression (NMS). So, we leverage the latest state-of-the-art object detection model, DINO, as a new graphical page object detector to detect tables, figures, and displayed formulas from document images.

Our DINO based graphical page object detection model consists of a Transformer encoder and a Transformer decoder. The Transformer encoder takes multi-scale feature maps $\{C_3, C_4, C_5, C_6\}$ output by the shared CNN backbone as input and generates a manageable number of region proposals for graphical page objects, whose bounding boxes are used to initialize the positional embeddings of object queries. The Transformer decoder takes object queries as input and outputs the final set of predictions in parallel. To reduce computation cost, the deformable attention mechanism [54] is adopted in the self attention layers in the encoder and cross attention layers in the decoder, respectively. To speed up model convergence, a contrastive denoising based training method for object queries is used. We refer readers to [49] for more details. Experimental results demonstrate that this new model can achieve superior graphical page object detection performance on PubLayNet and DocLayNet benchmark datasets.

3.3 Bottom-Up Text Region Detection Model

A text region is a semantic unit of writing consisting of a group of text-lines arranged in natural reading order and associated with a logical label, such as paragraph, list/list item, title, section heading, header, footer, footnote, and caption. Given a document image D composed of n text-lines $[t_1, t_2, ..., t_n]$, the goal of our bottom-up text region detection model is to group these text-lines into different text regions according to reading order and recognize the logical role of each text region. In this work, we assume the bounding boxes and textual contents of text-lines have already been given by a PDF parser or OCR engine. Based on the detection results of our DINO based graphical page object detection model, we first filter out those text-lines located inside graphical page objects and

then take the remaining text-lines as input. As shown in Fig. 2, our bottom-up text region detection model consists of a multi-modal feature extraction module, a multi-modal feature enhancement module, and two prediction heads, i.e., a reading order relation prediction head and a logical role classification head. The detailed illustrations of the multi-modal feature enhancement module and two prediction heads could be found in Fig. 3.

Multi-modal Feature Extraction Module. In this module, we extract the visual embedding, text embedding, and 2D position embedding for each text-line.

Visual Embedding. As shown in Fig. 2, we first resize C_4 and C_5 to the size of C_3 and then concatenate these three feature maps along the channel axis, which are fed into a 3×3 convolutional layer to generate a feature map C_{fuse} with 64 channels. For each text-line t_i, we adopt RoIAlign algorithm [15] to extract 7×7 features from C_{fuse} based on its bounding box $b_i = (x_i^1, y_i^1, x_i^2, y_i^2)$, where (x_i^1, y_i^1), (x_i^2, y_i^2) represent the coordinates of its top-left and bottom-right corners respectively. The final visual embedding V_i of t_i can be represented as:

$$V_i = LN(ReLU(FC(ROIAlign(C_{fuse}, b_i)))), \tag{1}$$

where FC is a fully-connected layer with 1,024 nodes and LN represents Layer Normalization [1].

Text Embedding. We leverage the pre-trained language model BERT [9] to extract the text embedding of each text-line. Specifically, we first serialize all the text-lines in a document image into a 1D sequence by reading them in a top-left to bottom-right order and tokenize the text-line sequence into a sub-word token sequence, which is then fed into BERT to get the embedding of each token. After that, we average the embeddings of all the tokens in each text-line t_i to obtain its text embedding T_i, followed by a fully-connected layer with 1,024 nodes to make the dimension the same as that of V_i:

$$T_i = LN(ReLU(FC(T_i))) \ . \tag{2}$$

2D Position Embedding. For each text-line t_i, we encode its bounding box and size information as its 2D position embedding B_i:

$$B_i = LN(MLP(x_i^1/W, y_i^1/H, x_i^2/W, y_i^2/H, w_i/W, h_i/H)), \tag{3}$$

where (w_i, h_i) and (W, H) represent the width and height of b_i and the input image, respectively. MLP consists of 2 fully-connected layers with 1,024 nodes, each of which is followed by ReLU.

For each text-line t_i, we concatenate its visual embeddings V_i, text embeddings T_i, and 2D position embeddings B_i to obtain its **multi-modal representation** U_i.

$$U_i = FC(Concat(V_i, T_i, B_i)), \tag{4}$$

where FC is a fully-connected layers with 1,024 nodes.

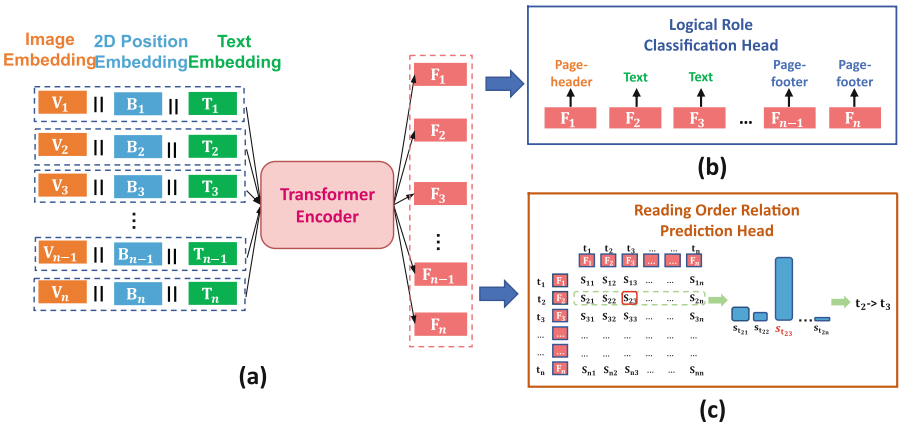

Fig. 3. Illustration of (a): Multi-modal Feature Enhancement Module; (b): Logical Role Classification Head; (c): Reading Order Relation Prediction Head.

Multi-modal Feature Enhancement Module. As shown in Fig. 3, we use a lightweight Transformer encoder to further enhance the multi-modal representations of text-lines by modeling their interactions with the self-attention mechanism. Each text-line is treated as a token of the Transformer encoder and its multi-modal representation is taken as the input embedding:

$$F = TransformerEncoder(U) \tag{5}$$

where $U = [U_1, U_2, ..., U_n]$ and $F = [F_1, F_2, ..., F_n]$ are the input and output embeddings of the Transformer encoder, n is the number of the input text-lines. To save computation, here we only use a 1-layer Transformer encoder, where the head number, dimension of hidden state, and the dimension of feedforward network are set as 12, 768, and 2048, respectively.

Reading Order Relation Prediction Head. We propose to use a relation prediction head to predict reading order relationships between text-lines. Given a text-line t_i, if a text-line t_j is its succeeding text-line in the same text region, we define that there exists a reading order relationship $(t_i \rightarrow t_j)$ pointing from text-line t_i to text-line t_j. If text-line t_i is the last (or only) text-line in a text region, its succeeding text-line is considered to be itself. Unlike many previous methods that consider relation prediction as a binary classification task [21,43], we treat relation prediction as a dependency parsing task and use a softmax cross entropy loss to replace the standard binary cross entropy loss during optimization by following [52]. Moreover, we adopt a spatial compatibility feature introduced in [50] to effectively model the spatial interactions between text-lines for relation prediction.

Specifically, we use a multi-class (i.e., n-class) classifier to calculate a score s_{ij} to estimate how likely t_j is the succeeding text-line of t_i as follows:

$$f_{ij} = FC_q(F_i) \circ FC_k(F_j) + MLP(r_{b_i,b_j}), \tag{6}$$

$$s_{ij} = \frac{\exp(f_{ij})}{\sum_N \exp(f_{ij})}, \tag{7}$$

where each of FC_q and FC_k is a single fully-connected layer with 2,048 nodes to map F_i and F_j into different feature spaces; $\circ$ denotes dot product operation; MLP consists of 2 fully-connected layers with 1,024 nodes and 1 node respectively; r_{b_i,b_j} is a spatial compatibility feature vector between b_i and b_j, which is a concatenation of three 6-d vectors:

$$r_{b_i,b_j} = (\Delta(b_i, b_j), \Delta(b_i, b_{ij}), \Delta(b_j, b_{ij})), \tag{8}$$

where b_{ij} is the union bounding box of b_i and b_j; $\Delta(.,.)$ represents the box delta between any two bounding boxes. Taking $\Delta(b_i, b_j)$ as an example, $\Delta(b_i, b_j) = (d_{ij}^{x_{\text{ctr}}}, d_{ij}^{y_{\text{ctr}}}, d_{ij}^{w}, d_{ij}^{h}, d_{ji}^{x_{\text{ctr}}}, d_{ji}^{y_{\text{ctr}}})$, where each dimension is given by:

$$\begin{aligned} d_{ij}^{x_{\text{ctr}}} &= (x_i^{\text{ctr}} - x_j^{\text{ctr}})/w_i, & d_{ij}^{y_{\text{ctr}}} &= (y_i^{\text{ctr}} - y_j^{\text{ctr}})/h_i, \\ d_{ij}^{w} &= \log(w_i/w_j), & d_{ij}^{h} &= \log(h_i/h_j), \\ d_{ji}^{x_{\text{ctr}}} &= (x_j^{\text{ctr}} - x_i^{\text{ctr}})/w_j, & d_{ji}^{y_{\text{ctr}}} &= (y_j^{\text{ctr}} - y_i^{\text{ctr}})/h_j, \end{aligned} \tag{9}$$

where $(x_i^{\text{ctr}}, y_i^{\text{ctr}})$ and $(x_j^{\text{ctr}}, y_j^{\text{ctr}})$ are the center coordinates of b_i and b_j, respectively.

We select the highest score from scores $[s_{ij}, k = 1, 2, ..., n]$ and output the corresponding text-line as the succeeding text-line of t_i. To achieve a higher relation prediction accuracy, we use another relation prediction head to identify the preceding text-line for each text-line further. The relation prediction results from both heads are combined to obtain the final results.

Logical Role Classification Head. Given the enhanced multi-modal representations of text-lines $F = [F_1, F_2, ..., F_n]$, we add a multi-class classifier to predict a logical role label for each text-line.

3.4 Optimization

Loss for DINO-Based Graphical Page Object Detection Model. The loss function of our DINO-based graphical page object detection model $L_{graphical}$ is exactly the same as L_{DINO} used in DINO [49], which is composed of multiple bounding box regression losses and classification losses derived from prediction heads and denoising heads. The bounding box regression loss is a linear combination of the L_1 loss and the GIoU loss [7], while the classification loss is the focal loss [25]. We refer readers to [49] for more details.

Loss for Bottom-up Text Region Detection Model. There are two prediction heads in our bottom-up text region detection model, i.e., a reading order relation prediction head and a logical role classification head. For reading order relation prediction, we adopt a softmax cross-entropy loss as follows:

$$L_{relation} = \frac{1}{N} \sum_{i} L_{\text{CE}} \left(\boldsymbol{s}_i, \boldsymbol{s}_i^* \right) \tag{10}$$

where $s_i = [s_{i1}, s_{i2}, ..., s_{iN}]$ is the predicted relation score vector calculated by Eqs. (6)–(7) and s_i^* is the target label.

We also adopt a softmax cross-entropy loss for logical role classification, which can be defined as

$$L_{role} = \frac{1}{N} \sum_{i} L_{\text{CE}} \left(\boldsymbol{c}_i, \boldsymbol{c}_i^* \right) \tag{11}$$

where c_i is the predicted label of the i^{th} text-line output by a softmax function and c_i^* is the corresponding ground-truth label.

Overall Loss. All the components in our approach are jointly trained in an end-to-end manner. The overall loss is the sum of $L_{graphical}$, $L_{relation}$ and L_{role}:

$$L_{overall} = L_{graphical} + L_{relation} + L_{role} \ . \tag{12}$$

4 Experiments

4.1 Datasets and Evaluation Protocols

We conduct experiments on two representative benchmark datasets, i.e., PubLayNet [53] and DocLayNet [36] to verify the effectiveness of our approach.

PubLayNet [53] is a large-scale dataset for document layout analysis released by IBM, which contains 340,391, 11,858, and 11,983 document pages for training, validation, and testing, respectively. All the documents in this dataset are scientific papers publicly available on PubMed Central and all the ground-truths are automatically generated by matching the XML representations and the content of corresponding PDF files. It pre-defines 5 types of page objects, including Text (i.e., Paragraph), Title, List, Figure, and Table. The summary of this dataset is shown in the left part of Table 1. Because ground-truths of the testing set are not publicly available, we evaluate our approach on the validation dataset.

DocLayNet [36] is a challenging human-annotated document layout analysis dataset newly released by IBM, which contains 69,375, 6,489, and 4,999 document pages for training, testing, and validation, respectively. It covers a variety of document categories, including Financial reports, Patents, Manuals, Laws, Tenders, and Scientific Papers. It pre-defines 11 types of page objects, including Caption, Footnote, Formula, List-item, Page-footer, Page-header, Picture, Section-header, Table, Text (i.e., Paragraph), and Title. The summary of this dataset is shown in the right part of Table 1.

Table 1. Summary of PubLayNet and DocLayNet datasets.

PublayNet			DocLayNet			
Page Object	Training	Validation	Page Object	Training	Testing	Validation
Text	2,343,356	88,625	Text	431,223	49,186	29,939
Title	627,125	18,801	Title	4,424	299	332
List	80,759	4239	List-item	161,779	13,320	10,525
Figure	109,292	4327	Picture	39,621	2,775	3,533
Table	102,514	4769	Table	300,116	2,269	2,395
–	–	–	Caption	19,199	1,763	1,544
–	–	–	Footnote	5,647	312	386
–	–	–	Formula	21,175	1,894	1,969
–	–	–	Page-footer	61,267	5,571	3,992
–	–	–	Page-header	47,997	6,683	3,366
–	–	–	Section-header	118,581	15,744	8,549
Image Page	340,391	11,858	Image Page	69,375	6,489	4,999

In addition to document images, these two datasets also provide corresponding original PDF files. Therefore, we can directly use a PDF parser (e.g., PDFMiner) to obtain the bounding boxes, text contents, and reading order of textlines for exploring our bottom-up text region detection approach. The evaluation metric of these two datasets is the COCO-style mean average precision (mAP) at multiple intersection over union (IoU) thresholds between 0.50 and 0.95 with a step of 0.05.

4.2 Implementation Details

We implement our approach based on Pytorch v1.10 and all experiments are conducted on a workstation with 8 Nvidia Tesla V100 GPUs (32 GB memory). Note that, on PubLayNet, a list refers to a whole list object consisting of multiple list items, whose label is not consistent with that of a text or a title. To reduce ambiguity, we consider all lists as specific graphical page objects and use our DINO based graphical page object detection model to detect them. In training, the parameters of the CNN backbone network are initialized with a ResNet-50 model [16] pre-trained on the ImageNet classification task, while the parameters of the text embedding extractor in our bottom-up text region detection model are initialized with the pre-trained $\text{BERT}_{\text{BASE}}$ model [9]. The parameters of the newly added layers are initialized by using random weights with a Gaussian distribution of mean 0 and standard deviation 0.01. The models are optimized by AdamW [31] algorithm with a batch size of 16 and trained for 12 epochs on PubLayNet and 24 epochs on DocLayNet. The learning rate and weight decay are set as 1e-5 and 1e-4 for the CNN backbone network, 2e-5 and 1e-2 for $\text{BERT}_{\text{BASE}}$, and 1e-4 and 1e-4 for the newly added layers, respectively. The

Table 2. Ablation studies on DocLayNet testing set (in %).

	DINO	Hybrid (V)	Hybrid (V+BERT-3L)	Hybrid (V+BERT-12L)
Caption	85.5	83.2	81.9	83.2
Footnote	69.2	67.8	68.5	69.7
Formula	63.8	63.9	64.2	63.4
List-item	80.9	86.9	87.5	**88.6**
Page-footer	54.2	89.9	86.1	90.0
Page-header	63.7	70.4	**81.3**	76.3
Picture	84.1	82.0	81.8	81.6
Section-header	64.3	86.2	84.2	83.2
Table	85.7	84.4	85.4	84.8
Text	83.3	86.1	85.6	84.8
Title	82.8	75.3	82.3	**84.9**
mAP	74.3	79.6	80.8	**81.0**

Table 3. Ablation studies on PubLayNet validation set (in %).

Method	Text	Title	List	Table	Figure	mAP
DINO	94.9	91.4	96.0	98.0	97.3	95.52
Hybrid (V)	97.0	92.8	**96.4**	98.1	**97.4**	96.34
Hybrid (V+BERT-3L)	**97.7**	93.1	96.3	98.1	97.2	96.48
Hybrid (V+BERT-12L)	97.4	**93.5**	**96.4**	**98.2**	97.2	**96.54**

learning rate is divided by 10 at the 11^{th} epoch for PubLayNet and 20^{th} epoch for DocLayNet. The other hyper-parameters of AdamW including betas and epsilon are set as (0.9, 0.999) and 1e-8. We also adopt a multi-scale training strategy. The shorter side of each image is randomly rescaled to a length selected from [512, 640, 768], while the longer side should not exceed 800.

In the testing phase, we set the shorter side of the input image as 640. We group text-lines into text regions based on predicted reading order relationships by using the Union-Find algorithm. The logical role of a text region is determined by majority voting, and the bounding box is the union bounding box of all its constituent text-lines.

4.3 Ablation Studies

Effectiveness of Hybrid Strategy. In this section, we evaluate the effectiveness of the proposed hybrid strategy. To this end, we train two baseline models: 1) A DINO baseline to detect both graphical page objects and text regions; 2) A hybrid model (denoted as Hybrid(V)) that only uses visual and 2D position features for bottom-up text region detection. As shown in the first two columns of Table 2, compared with the DINO model, the Hybrid(V) model

can achieve comparable graphical page object detection results but much higher text region detection accuracy on DocLayNet, leading to a 5.3% improvement in terms of mAP. In particular, the Hybrid(V) model can significantly improve small-scale text region detection performance, e.g., 89.9% vs. 54.2% for Page-footer, 70.4% vs. 63.7% for Page-header and 86.2% vs. 64.3% for Section-header. We observe that this accuracy improvement is mainly owing to the higher localization accuracy. Experimental results on PubLayNet are listed in the first two rows of Table 3. We can see that the Hybrid(V) model improves AP by 2.1% for Text and 1.4% for Title, leading to a 0.82% improvement in terms of mAP on PubLayNet. These experimental results clearly demonstrate the effectiveness of the proposed hybrid strategy that combines the best of both top-down and bottom-up methods.

Effectiveness of Using Textual Features. In order to evaluate the effectiveness of using textual features, we compare the performance of three hybrid models, i.e., Hybrid(V), Hybrid(V+BERT-3L) and Hybrid(V+BERT-12L). The bottom-up text region detection model of Hybrid(V) does not use textual features, while the models of Hybrid(V+BERT-3L) and Hybrid(V+BERT-12L) use the first 3 and 12 Transformer blocks of $\text{BERT}_{\text{BASE}}$ to extract text embeddings for text-lines, respectively. Experimental results on DocLayNet and PubLayNet are listed in the second and third parts of Table 2 and Table 3. We can see that both Hybrid(V+BERT-3L) and Hybrid(V+BERT-12L) models are consistently better than Hybrid(V), and Hybrid(V+BERT-12L) achieves the best results. The large performance improvements of these two models mainly come from the categories of Title, Page-header and List-item on DocLayNet and the categories of Text and Title on PubLayNet, respectively.

Table 4. Performance comparisons on DocLayNet testing set (in %). The results of Mask R-CNN, Faster R-CNN, and YOLOv5 are obtained from [36].

	Human	Mask R-CNN	Faster R-CNN	YOLOv5	DINO	Proposed
Caption	84-89	71.5	70.1	77.7	**85.5**	83.2
Footnote	83-91	71.8	73.7	**77.2**	69.2	69.7
Formula	83-85	63.4	63.5	**66.2**	63.8	63.4
List-item	87-88	80.8	81.0	86.2	80.9	**88.6**
Page-footer	93-94	59.3	58.9	61.1	54.2	**90.0**
Page-header	85-89	70.0	72.0	67.9	63.7	**76.3**
Picture	69-71	72.7	72.0	77.1	**84.1**	81.6
Section-header	83-84	69.3	68.4	74.6	64.3	83.2
Table	77-81	82.9	82.2	**86.3**	85.7	84.8
Text	84-86	85.8	85.4	**88.1**	83.3	84.8
Title	60-72	80.4	79.9	82.7	82.8	**84.9**
mAP	82-83	73.5	73.4	76.8	74.3	**81.0**

Table 5. Performance comparisons on PubLayNet validation set (in %). Vision and Text stands for using visual and textual features, respectively.

Method	Modality	Text	Title	List	Table	Figure	mAP
Faster R-CNN [53]	Vision	91.0	82.6	88.3	95.4	93.7	90.2
Mask R-CNN [53]		91.6	84.0	88.6	96.0	94.9	91.0
Naik et al. [34]		94.3	88.7	94.3	97.6	96.1	94.2
Minouei et al. [33]		94.4	90.8	94.0	97.4	96.6	94.6
DiT-L [20]		94.4	89.3	96.0	97.8	**97.2**	94.9
SRRV [2]		95.8	90.1	95.0	97.6	96.7	95.0
DINO [49]		94.9	91.4	96.0	98.0	97.3	95.5
TRDLU [46]		95.8	92.1	97.6	97.6	96.6	96.0
UDoc [13]	Vision+Text	93.9	88.5	93.7	97.3	96.4	93.9
LayoutLMv3 [17]		94.5	90.6	95.5	97.9	97.0	95.1
VSR [51]		96.7	93.1	94.7	97.4	96.4	95.7
Proposed	Vision	97.0	92.8	96.4	98.1	**97.4**	96.3
Proposed	Vision+Text	**97.4**	**93.5**	**96.4**	**98.2**	97.2	**96.5**

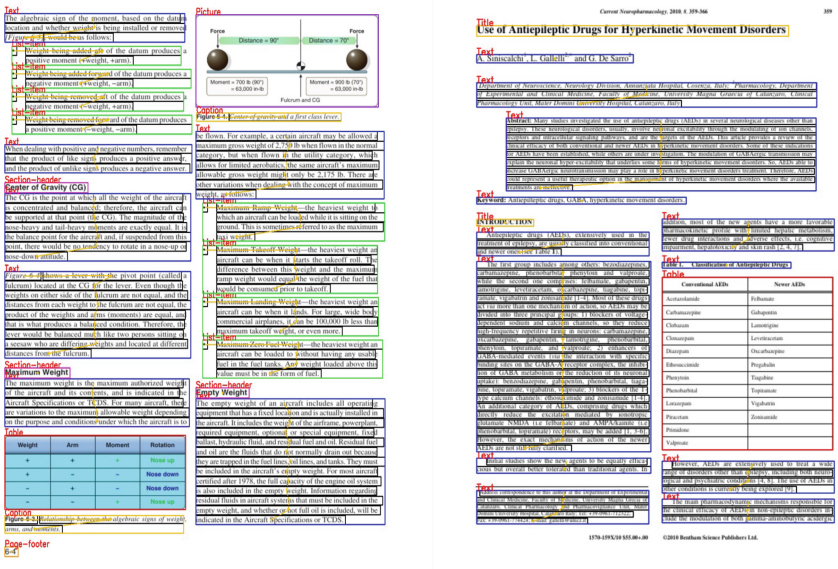

Fig. 4. Qualitative results of our approach: DocLayNet (Left); PubLayNet (Right).

4.4 Comparisons with Prior Arts

DocLayNet. We compare our approach with other most competitive methods, including Mask R-CNN, Faster R-CNN, YOLOv5, and DINO on DocLayNet. As shown in Table 4, our approach outperforms the closest method YOLOv5 substantially by improving mAP from 76.8% to 81.0%. Considering that DocLayNet

is an extremely challenging dataset that covers a variety of document scenarios and contains a large number of text regions with fine-grained logical roles, the superior performance achieved by our proposed approach can demonstrate the advantage of our approach.

PubLayNet. We compare our approach with several state-of-the-art vision-based and multi-modal methods on PubLayNet. Experimental results are presented in Table 5. We can see that our approach outperforms all these methods no matter whether textual features are used in our bottom-up text region detection model or not.

Qualitative Results. The state-of-the-art performance achieved on these two datasets demonstrates the effectiveness and robustness of our approach. Furthermore, our approach provides a new capability of outputting the reading order of text-lines in each text region. Some qualitative results are depicted in Fig. 4.

5 Summary

In this paper, we propose a new hybrid document layout analysis approach, which consists of a new DINO based graphical page object detection model to detect tables, figures, and formulas in a top-down manner and a new bottom-up text region detection model to group text-lines located outside graphical page objects into text regions according to reading order and recognize the logical role of each text region by using both visual and textual features. The state-of-the-art results obtained on DocLayNet and PubLayNet demonstrate the effectiveness and superiority of our approach. Furthermore, in addition to bounding boxes, our approach can also output the reading order of text-lines in each text region directly, which is crucial to various downstream tasks such as translation and information extraction. In future work, we will explore how to use a unified model to solve various physical and logical layout analysis tasks, including page object detection, inter-object relation prediction, list parsing, table of contents generation, and so on.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Bi, H., et al.: Srrv: A novel document object detector based on spatial-related relation and vision. IEEE Transactions on Multimedia (2022)
3. Binmakhashen, G.M., Mahmoud, S.A.: Document layout analysis: a comprehensive survey. ACM Comput. Surv. (CSUR) **52**(6), 1–36 (2019)
4. Biswas, S., Banerjee, A., Lladós, J., Pal, U.: Docsegtr: an instance-level end-to-end document image segmentation transformer. arXiv preprint arXiv:2201.11438 (2022)

5. Cai, Z., Vasconcelos, N.: Cascade r-cnn: high quality object detection and instance segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **43**(5), 1483–1498 (2019)
6. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision, pp. 213–229 (2020)
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision, pp. 213–229 (2020)
8. Dai, X., et al.: Dynamic head: Unifying object detection heads with attentions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7373–7382 (2021)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186 (2019)
10. Doermann, D., Tombre, K. (eds.): Handbook of Document Image Processing and Recognition. Springer, London (2014). https://doi.org/10.1007/978-0-85729-859-1
11. Girshick, R.: Fast r-cnn. In: Proceedings of the International Conference on Computer Vision, pp. 1440–1448 (2015)
12. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
13. Gu, J., et al.: Unified pretraining framework for document understanding. arXiv preprint arXiv:2204.10939 (2022)
14. He, D., Cohen, S., Price, B., Kifer, D., Giles, C.L.: Multi-scale multi-task fcn for semantic page segmentation and table detection. In: Proceedings of the International Conference on Document Analysis and Recognition. vol. 1, pp. 254–261 (2017)
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-CNN. In: Proceedings of the International Conference on Computer Visio, pp. 2961–2969 (2017)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
17. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: Layoutlmv3: Pre-training for document ai with unified text and image masking. In: Proceedings of the ACM International Conference on Multimedia, pp. 4083–4091 (2022)
18. Jocher, G., et al.: ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations (Apr 2021)
19. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. arXiv preprint arXiv:2203.01305 (2022)
20. Li, J., Xu, Y., Lv, T., Cui, L., Zhang, C., Wei, F.: Dit: Self-supervised pre-training for document image transformer. In: Proceedings of the ACM International Conference on Multimedia. pp. 3530–3539 (2022)
21. Li, X.H., Yin, F., Liu, C.L.: Page object detection from pdf document images by deep structured prediction and supervised clustering. In: Proceedings of the International Conference on Pattern Recognition, pp. 3627–3632 (2018)
22. Li, X.H., Yin, F., Liu, C.L.: Page segmentation using convolutional neural network and graphical model. In: Proceedings of the International Workshop on Document Analysis Systems, pp. 231–245 (2020)

23. Li, X.H., et al.: Instance aware document image segmentation using label pyramid networks and deep watershed transformation. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 514–519 (2019)
24. Li, Y., Zou, Y., Ma, J.: Deeplayout: A semantic segmentation approach to page layout analysis. In: Proceedings of the International Conference on Intelligent Computing Methodologies, pp. 266–277 (2018)
25. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the International Conference on Computer Vision, pp. 2980–2988 (2017)
26. Liu, S., et al.: Dab-detr: Dynamic anchor boxes are better queries for detr. arXiv preprint arXiv:2201.12329 (2022)
27. Liu, S., Wang, R., Raptis, M., Fujii, Y.: Unified line and paragraph detection by graph convolutional networks. In: Proceedings of the International Workshop on Document Analysis Systems, pp. 33–47 (2022)
28. Liu, Z., et al.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 10012–10022 (2021)
29. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
30. Long, S., Qin, S., Panteleev, D., Bissacco, A., Fujii, Y., Raptis, M.: Towards end-to-end unified scene text detection and layout analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1049–1059 (2022)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
32. Luo, S., Ding, Y., Long, S., Han, S.C., Poon, J.: Doc-gcn: Heterogeneous graph convolutional networks for document layout analysis. arXiv preprint arXiv:2208.10970 (2022)
33. Minouei, M., Soheili, M.R., Stricker, D.: Document layout analysis with an enhanced object detector. In: Proceedings of the International Conference on Pattern Recognition and Image Analysis, pp. 1–5 (2021)
34. Naik, S., Hashmi, K.A., Pagani, A., Liwicki, M., Stricker, D., Afzal, M.Z.: Investigating attention mechanism for page object detection in document images. Appl. Sci. **12**(15), 7486 (2022)
35. Oliveira, D.A.B., Viana, M.P.: Fast cnn-based document layout analysis. In: Proceedings of the International Conference on Computer Vision Workshops, pp. 1173–1180 (2017)
36. Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A.S., Staar, P.W.: Doclaynet: A large human-annotated dataset for document-layout analysis. arXiv preprint arXiv:2206.01062 (2022)
37. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 91–99 (2015)
38. Saha, R., Mondal, A., Jawahar, C.: Graphical object detection in document images. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 51–58 (2019)
39. Sang, Y., Zeng, Y., Liu, R., Yang, F., Yao, Z., Pan, Y.: Exploiting spatial attention and contextual information for document image segmentation. In: Proceedings of the Advances in Knowledge Discovery and Data Mining, pp. 261–274 (2022)
40. Shi, C., Xu, C., Bi, H., Cheng, Y., Li, Y., Zhang, H.: Lateral feature enhancement network for page object detection. IEEE Trans. Instrum. Meas. **71**, 1–10 (2022)

41. Sun, P., et al.: Sparse r-cnn: End-to-end object detection with learnable proposals. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 14454–14463 (2021)
42. Vo, N.D., Nguyen, K., Nguyen, T.V., Nguyen, K.: Ensemble of deep object detectors for page object detection. In: Proceedings of the International Conference on Ubiquitous Information Management and Communicatio, pp. 1–6 (2018)
43. Wang, R., Fujii, Y., Popat, A.C.: Post-ocr paragraph recognition by graph convolutional networks. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp. 493–502 (2022)
44. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: Solov2: Dynamic and fast instance segmentation. In: Proceedings of the Advances in Neural information processing systems. vol. 33, pp. 17721–17732 (2020)
45. Xue, C., Huang, J., Zhang, W., Lu, S., Wang, C., Bai, S.: Contextual text block detection towards scene text understanding. In: Proceedings of the European Conference on Computer Vision, pp. 374–391 (2022)
46. Yang, H., Hsu, W.: Transformer-based approach for document layout understanding. In: Proceedings of the International Conference on Image Processing, pp. 4043–4047 (2022)
47. Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., Lee Giles, C.: Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5315–5324 (2017)
48. Yi, X., Gao, L., Liao, Y., Zhang, X., Liu, R., Jiang, Z.: Cnn based page object detection in document images. In: Proceedings of the International Conference on Document Analysis and Recognition. vol. 1, pp. 230–235 (2017)
49. Zhang, H., et al.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)
50. Zhang, J., Elhoseiny, M., Cohen, S., Chang, W., Elgammal, A.: Relationship proposal networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5678–5686 (2017)
51. Zhang, P., et al.: Vsr: a unified framework for document layout analysis combining vision, semantics and relations. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 115–130 (2021)
52. Zhang, Y., Bo, Z., Wang, R., Cao, J., Li, C., Bao, Z.: Entity relation extraction as dependency parsing in visually rich documents. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 2759–2768 (2021)
53. Zhong, X., Tang, J., Yepes, A.J.: Publaynet: largest dataset ever for document layout analysis. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 1015–1022 (2019)
54. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: Proceedings of the International Conference on Learning Representations (2021)