Intelligent Document Parsing: Towards End-to-end Document Parsing via Decoupled Content Parsing and Layout Grounding

Hangdi Xing[1], **Feiyu Gao**[1*], **Qi Zheng**[1], **Zhaoqing Zhu**[1], **Zirui Shao**[2], **Ming Yan**[1]
[1]Alibaba Group [2]Zhejiang University
{hangdi.xhd, feiyu.gfy, ym119608}@alibaba-inc.com,
{zhengqisjtu,zzhaoqing.z}@gmail.com, shaozirui@zju.edu.cn

Abstract

In the daily work, vast amounts of documents are stored in pixel-based formats such as images and scanned PDFs, posing challenges for efficient database management and data processing. Existing methods often fragment the parsing process into the pipeline of separated subtasks on the layout element level, resulting in incomplete semantics and error propagation. Even though models based on multimodal large language models (MLLMs) mitigate the issues to some extent, they also suffer from absent or sub-optimal grounding ability for visual information. To address these challenges, we introduce the Intelligent Document Parsing (IDP) framework, an end-to-end document parsing framework leveraging the vision-language priors of MLLMs, equipped with an elaborately designed document representation and decoding mechanism to decouple the content parsing and layout grounding to fully activate the potential of MLLMs for document parsing. Experimental results demonstrate that the IDP method surpasses existing methods, significantly advancing MLLM-based document parsing.

1 Introduction

In contemporary settings, a substantial volume of information is produced daily and stored within pixel-based representations, such as images and scanned PDFs, rather than arranged in machine-understandable structured formats, such as JSON and HTML. This scenario presents considerable challenges in practice, as structured formats are indispensable for efficient database storage and standardized data processing (Johnson et al., 2003; Clifton and Garcia-Molina, 2000), as well as for downstream applications, including information retrieval and natural language processing (Wilkinson, 1994; Dasigi et al., 2021; Saad-Falcon et al., 2023; Mo et al., 2025).

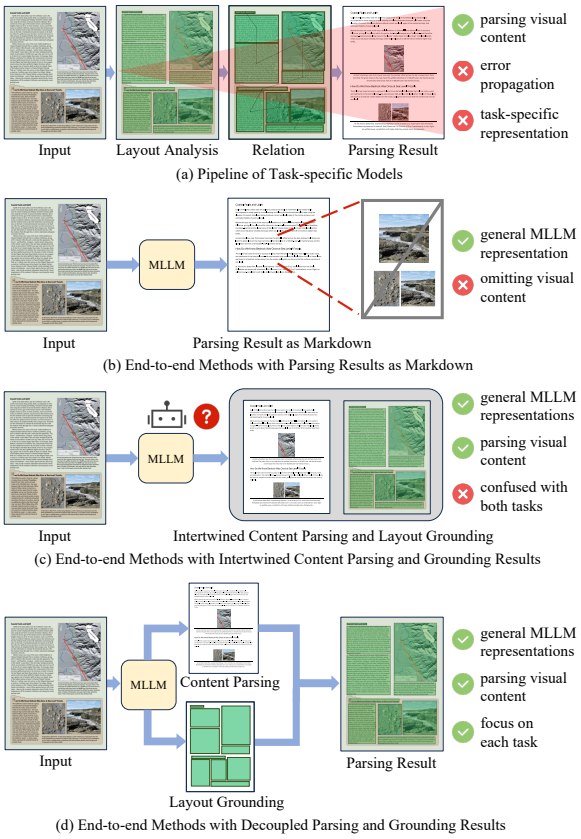


Figure 1: The comparison between different document parsing paradigms.

Document parsing (Yao, 2023; Zhang et al., 2024; Wang et al., 2024a) aims at extracting the content of unstructured documents. The task is challenging due to the diversity of the layout and logical structures (Li et al., 2024), incorporating a plethora of plain text, visual elements, such as images and icons, and multi-modal information, such as tables and charts. This diversity and complexity require document parsing technologies to possess strong adaptability and generalization capabilities to handle various types of documents effectively (Xing et al., 2024; Ouyang et al., 2024).

Most existing document parsing methods lever-

* Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 19987–19998

age the pipelines of separated subtasks, as shown in fig. 1 (a), including document layout analysis (Zhong et al., 2019; Cheng et al., 2023), reading order prediction (Wang et al., 2021), and document hierarchy parsing (Li et al., 2024; Xing et al., 2024). The main drawbacks of the pipeline methods are: 1) By encoding fragmented layout elements independently, these methods fail to capture the holistic information of documents, and thus result in semantic loss. 2) The pipeline framework suffers from the error propagation problem. Recently, some research (Wang et al., 2024b; Chen et al., 2025; Wang et al., 2025) is dedicated to designing end-to-end frameworks based on task-specific models to mitigate this issue. However, they are also based on local clues such as text lines, and without effective pre-training. Thus, they suffer from the over-segmented and ineffective semantic representations, especially when handling the sophisticated logical structure of documents.

There have also been attempts to apply advanced large language models (LLMs) to conduct end-to-end document parsing (Wei et al., 2024; Hu et al., 2024; Bai et al., 2025; Zhu et al., 2025a) as an image-to-markdown task, as shown in fig. 1 (b); however, the Markdown representation of these methods fails to explicitly capture the multi-modal elements such as images, icons and charts. Bai et al. (2025); Feng et al. (2025) introduce the layout grounding task to mitigate this issue, as shown in fig. 1 (c). But the method results in suboptimal performance in both content parsing and layout grounding tasks, due to the distractions between the two heterogeneous tasks (Rasheed et al., 2024; Surís et al., 2023; Jiang et al., 2024), significantly affecting the document parsing performance.

To address this issue, in this paper, we propose an end-to-end framework, Intelligent Document Parsing (IDP), which leverages the powerful vision-language priors of MLLMs to build an effective document parsing method that possesses exceeding effectiveness. As illustrated in fig. 1 (d), the IDP framework decouples the content parsing and layout grounding tasks, allowing different decoder modules to focus on the tasks they excel at. This approach fully unleashes the potential of MLLMs in the document parsing situation. Specifically, on one hand, we developed the IDP representation format, akin to HTML sequences and integrating information of layout elements, their reading order, and logical relationships. On the other hand, we addressed the issue of conflicts between content parsing and layout grounding by introducing the decoupled decoding mechanism. Experimental results show that this model outperforms leading MLLMs and shows advantages compared to pipelines composed of task-specific models.

Our main contributions can be summarized as follows:

- We propose the IDP, an end-to-end document parsing framework which effectively leverages the generalized vision-language priors of MLLMs to recognize and organize the text content of documents.
- We equip the model with a decoupled decoding mechanism to disentangle the content parsing and layout grounding tasks, which helps the MLLMs to focus on extracting and organizing the text content while enabling the framework to obtain multi-modal content.
- Experimental results validate the effectiveness of the IDP framework. The model shows obvious performance boost and provides a significant baseline for parsing real-world documents.

2 Related Work

2.1 Document AI

Document AI involves automated reading, extracting information from documents, and understanding documents with rich typesetting (Cui et al., 2021). As the world is going digital, it has received a heightened focus on its impact and significance (Sarkhel and Nandi, 2019, 2021; Zhu et al., 2025b; Shao et al., 2024). The document parsing subtasks (Zhang et al., 2024; Xing et al., 2023) refer to converting unstructured and semi-structured documents into structured information. Document parsing extracts elements like text, equations, tables, and images from various inputs while preserving their structural relationships. Document parsing is crucial for document understanding sub-tasks, reshaping how information is stored, shared, and applied across numerous applications.

2.2 Document Parsing Methods

Currently, most document parsing methods decompose the processings into multiple separated subtasks (mainly including document layout analysis (Zhong et al., 2019; Cheng et al., 2023; Wang et al., 2024a; Li et al., 2025), reading order prediction

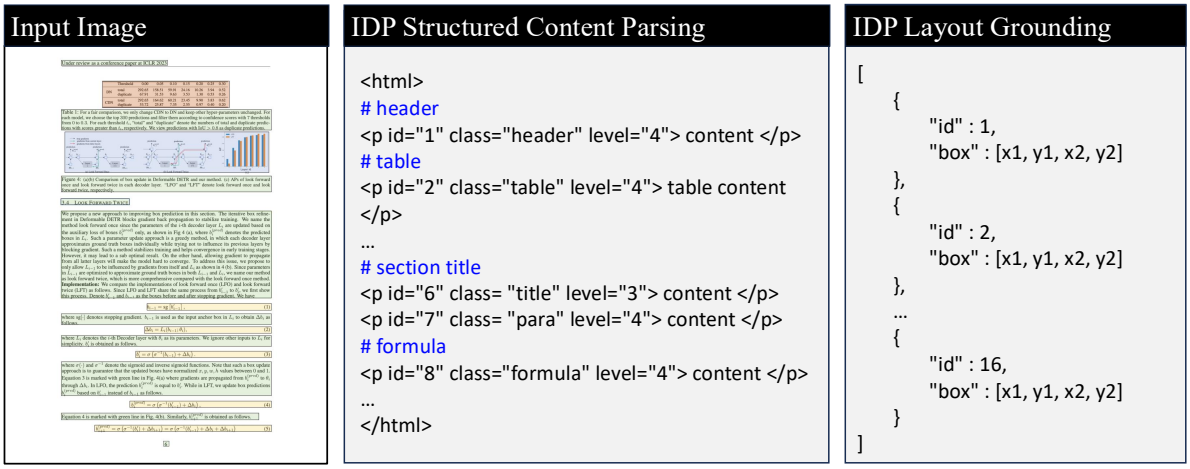


Figure 2: An overview of the proposed document representation IDP format, which decouples the content parsing and layout grounding results. The comments are for illustration only.

(Wang et al., 2021) and document hierarchy parsing (Li et al., 2024; Xing et al., 2024)), and then compose the components of different subtasks in a pipeline to predict the final structured machine-understandable document. The main drawbacks of these methods are that the pipeline framework suffers from the error propagation problem and fails to leverage the powerful visual-language models (Bai et al., 2025; Zhu et al., 2025a). Some researchers (Wang et al., 2024b; Chen et al., 2025; Wang et al., 2025) are dedicated to designing an end-to-end document parsing framework to mitigate this issue. However, they also suffer from the over-segmented and ineffective semantic representations, especially when handling the sophisticated logical structure of documents.

2.3 MLLMs for Document AI

Application of MLLM in document AI is a rapidly growing research area driven by increasing industrial demand (Luo et al., 2024; Hu et al., 2024; Bai et al., 2025; Zhu et al., 2025a). Most of them are focused on Document Question Answering (Mathew et al., 2021) and Document Information Extraction (Jaume et al., 2019) tasks, which achieve remarkable performance boosts of application systems. The extensive knowledge of large language models is highly beneficial for advanced understanding tasks in document parsing, such as determining the reading order and logical relationships of different sections. However, the capabilities of large language models in document parsing have not been fully exploited. Most existing work (Blecher et al., 2023; Wei et al., 2024) treats document parsing as markdown sequence generation, which has limited expressive power for capturing document hierarchy and multi-modal information. Bai et al. (2025) design a HTML-like document representation to mitigate this issue. However, the constrained grounding ability of the sequence decoder results in sub-optimal performance.

3 IDP Document Rerpresentation

Previous LLM-based document parsing methods (Blecher et al., 2023; Wei et al., 2024) often follow the markdown generation paradigm. However, important multi-modal information, such as images and charts, is omitted in this format. Therefore, in recent studies (Bai et al., 2025), researchers still prefer using parsing results that include bounding boxes of layout elements. In this paper, we consider the document parsing task as recognizing the layout elements and organizing them in an ordered and hierarchical structure. Specifically, the input is given as a document as image I. The output is the extracted layout elements $E = \{E_1, E_2, ..., E_M\}$ in traversal order, along with their hierarchical depth predicted. Each layout elements E_i is represented by the corresponding bounding boxes $B_i = [x_{i1}, y_{i1}, x_{i2}, y_{i2}]$, categories c_i, depth l_i in the document structure tree and the contents.

In order to empower an end-to-end model with comprehensive capabilities for parsing and extracting both text and multi-modal content from documents, we propose the document representation of IDP format, as shown in fig. 2. It decomposes the content-inferred task and the visual grounding task. Specifically, the layout elements are uniformly for-

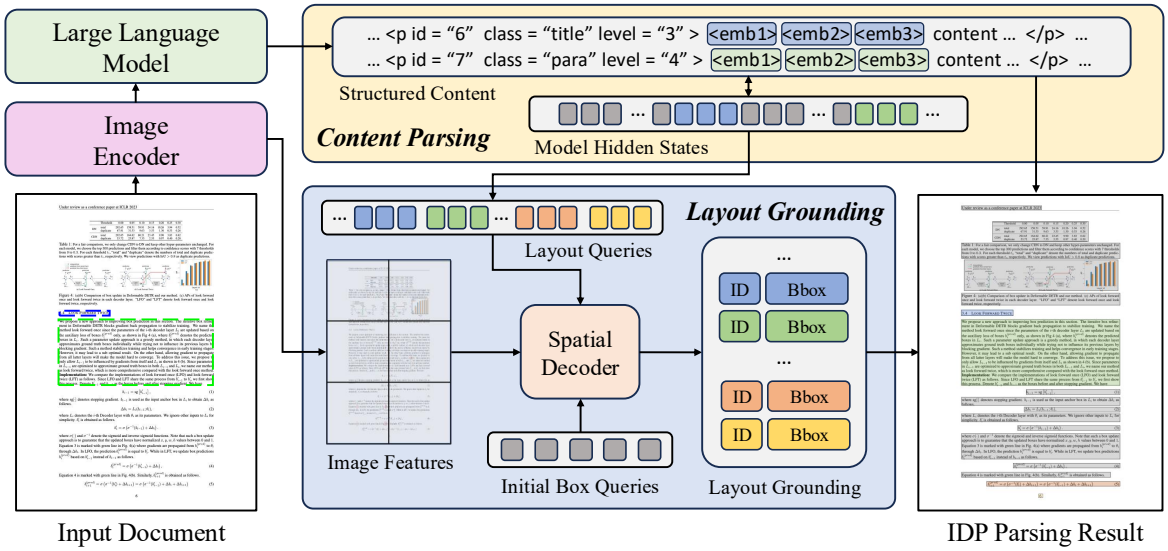


Figure 3: The illustration of the proposed MLLM-based document parsing framework. The LLM is focused on the content parsing task. The decoupled layout grounding module is triggered when a layout element is completely parsed to locate the element.

matted as HTML tags, which integrates reading orders (sequential order of tags), categories (attribute 'class'), and hierarchical depth (attribute 'level'). The tag contents of textual layout elements are the extracted text, while those of multi-modal elements are corresponding place holders. The positional information of layout elements is stored in an auxiliary data item, which is a list of bounding boxes. The correspondence between layout elements and bounding boxes is maintained through a one-to-one mapping based on unique IDs.

In this way, the IDP format caters better to the prior knowledge of MLLMs. It addresses the challenge faced by the markdown format in preserving multi-modal information. Additionally, it avoids coupling layout grounding results within the prediction sequence, thereby allowing MLLM to focus more effectively on document content parsing and organization.

4 IDP Model

In order to avoid conflicts among content-inferred and visual-spatial tasks (Zhu et al., 2022; Wu et al., 2024) and fully activate the potential of MLLMs, we propose a decoupled decoding mechanism to disentangle the task into two simultaneous sequential and spatial decoding processes. The overall architecture of the IDP model is depicted in fig. 3. It consists of 3 parts: (1) an image encoder for encoding the document images; (2) an LLM that models the multi-modal inputs and generates document contents organized in HTML; (3) a spatial decoder for the layout grounding task. In the decoding process, the layout grounding process is triggered by a generated layout element by the LLM, and it predicts grounding results based on layout queries from the LLM. The following subsections specify these crucial components and the decoding process, respectively.

4.1 Model Architecture

4.1.1 Image Encoder

To equip the IDP model with sufficient perception capabilities, we adopt an efficient high-resolution vision encoder for image encoding following Bai et al. (2025), which incorporates 2D-RoPE and window attention to support high-resolution input while accelerating the computation.

4.1.2 LLM Decoder

We use the LLM component of Qwen2.5-VL-3B Bai et al. (2025) for its basic document parsing ability and the compatibility with the image encoder. The LLM decoder is responsible for extracting and organizing content in the document. For multi-modal information, such as images and charts, it predicts the corresponding placeholders to properly activate the spatial decoder.

4.1.3 Spatial Decoder

The spatial decoder D_s is designed to predict the layout grounding of document parsing. The spatial decoder is a DETR-like framework based on

enhanced object queries from the LLM (Liu et al., 2024) to obtain the corresponding boxes for the layout elements generated by the LLM.

4.2 Decoupled Decoding Mechanism

In the decoupled decoding process, the LLM decoder focuses on parsing, sorting, and structuring document content. It also generates the query representations for layout grounding. When a layout element is completely parsed, the layout grounding process is activated, and the spatial decoder predicts the grounding results based on layout queries from LLM. The layout queries and the prediction of grounding results are detailed in the following sections.

4.2.1 Layout Element Quries

We extend the vocabulary of LLM by incorporating specialized tokens for pooling, including k query tokens <emb-0>, <emb-1>, ..., <emb-k>. Whenever the LLM predicts the opening tag of a layout element, the query tokens would be automatically appended after the predicted tokens. After the prediction of the endding tag, the last-layer hidden states $H_q \in R^{k \times d}$ of the layout queries are extracted and passed into an MLP projection to obtain $h^q \in R^{d'}$, where d and d' are the hidden size of the LLM and the spatial decoder respectively. Finally, h^q is sent into the spatial decoders as a condition to perform the grounding tasks.

4.2.2 Layout Grounding Prediction

The spatial decoder is activated when the layout element queries come, which are pooled from the LLM hidden states of the generated layout elements. Then the spatial decoder enhances its initial box queries following (Liu et al., 2024) and predicts the bounding box. To be specific, it predicts the corresponding boxes for layout element queries as :

$$\{B_i\}_{1,2,...,n} = \{D_s(I, h_i^q)\}_{1,2,...,n} \quad (1)$$

4.3 Training Strategies

To create a generalist model capable of parsing documents of various layout and structure, we propose a three-stage training strategy. The first stage focuses on building an MLLM with a strong awareness of content-oriented document parsing results. The subsequent stages aim to develop layout grounding capabilities for the IDP model while ensuring its content parsing abilities remain unaffected.

Stage 1: Multi-modal Training. In the first stage, we follow the typical instruction tuning settings with the LLM unfreezed on about 210,000 elaborately annotated document parsing data of paired visual documents and textual parsing results. This phase aims to establish the vision-language alignment on document images and enhance the content extraction and organization capability of the foundation MLLMs.

Stage 2: Joint Training of Decoders. At this stage, we integrate the spatial decoder into the model and perform multi-task joint training. During this stage, both the LLM decoder and the spatial decoder undergo one epoch of training on 80,000 manually annotated document parsing data with IDP format. In this way, the model learns to generate a structured document with the grounding results of layout elements.

Stage 3: Spatial decoder-only Fine-tuning. Since the spatial decoders cannot converge within a single epoch, we further train the decoders for three more epochs with an additional 100,000 synthesized data only for the grounding task. In this stage, the spatial decoder and query embeddings are trained while all other components are frozen.

5 Experiments

5.1 Implementation Details

5.1.1 Dataset

We conduct experiments on the following datasets: 1) HRDoc (Ma et al., 2023), which contains two subsets: HRDoc-Simple (HRDS) dataset with 1000 documents (10224 pages) from ACL and HRDoc-Hard (HRDH) dataset with 1500 documents (21427 pages) from arXiv. 2) DocHieNet (Xing et al., 2024), consisting of 1673 documents (15610 pages) from various types of documents in both English and Chinese. 3) OmniDocBench (Ouyang et al., 2024), including 981 pages from multiple sources of documents for zero-shot evaluation.

5.1.2 Model Details

We adopt the Qwen2.5-VL-3B (Bai et al., 2025) as the basic MLLM. A Grounding-DINO (Liu et al., 2024) with Swin-T (Liu et al., 2021) backbone is employed as the spatial decoder. All the components of Qwen2.5-VL are loaded from the pre-trained weights, while the spatial decoder is initialized from the pre-trained layout analysis model.

Method	Format	HRDoc-Simple			HRDoc-Hard			DocHieNet		
		G-F1	ARD	R-F1	G-F1	ARD	R-F1	G-F1	ARD	R-F1
GOT	md	-	1.80	83.21	-	2.54	70.54	-	4.83	57.44
InternVL3-8B	md	-	0.76	93.61	-	1.72	86.23	-	2.13	74.13
InternVL3-8B	grd	97.45	0.96	91.06	91.23	1.76	83.41	84.67	2.34	72.61
Qwen2.5-VL-7B	qwen	96.20	0.87	94.15	87.16	1.82	87.62	78.74	2.45	76.74
Qwen-VL-Max	-	61.90	1.04	91.23	56.52	1.89	84.80	59.84	2.52	64.32
GPT-4o	-	-	0.87	95.72	-	1.25	88.56	-	1.87	58.13
Ours	IDP	98.64	0.24	97.45	96.11	0.70	90.32	93.75	1.01	79.45

Table 1: Summary of performance of IDP model on different datasets compared to both open-source and closed-source baseline MLLMs. 'md', 'grd and 'qwen' refer to Markdown, InternVL-grounding and QwenVL-HTML format.

The IDP modal contains 3.2B parameters in general. The number k of query embeddings is set to 4. The dynamic resolution strategy is employed following Bai et al. (2025) with max pixel as 1,204,224. In stage 1, we adopt the AdamW optimizer (Loshchilov and Hutter, 2017) with the peak learning rate of 1e-3 and weight decay of 0. The training involves a total batch size of 16 across 8 A100 GPUs. In stage 2, we add the task-specific decoders and perform the multi-task fine-tuning. The image encoder and LLM decoder are trained with the peak learning rate of 1e-5, while the spatial decoder is trained with the peak learning rate of 1e-4. The model is also trained on 8 A100 GPUs with a batch size of 2 per GPU. In stage 3, we freeze all the components except for the spatial decoders and query embeddings to enhance the grounding capability of the IDP model. The model undergoes three training epochs on 8 A100 GPUs with a peak learning rate of 1e-4 and a total batch size of 32.

5.2 Evaluation Metric

The IDP model is responsible solely for layout grounding, ordering, and hierarchical structuring. Content parsing can be performed based on the results of the layout detection with external modules. So our evaluations primarily focus on layout-related metrics.

5.2.1 Layout Grounding

In task-specific layout analysis work (Zhong et al., 2019; Pfitzmann et al., 2022), the evaluation metric is the mean Average Precision (mAP). Since the grounding results based on LLM methods do not include confidence scores, we employ the F-1 with Intersection over Union (IoU) threshold of 0.75 as the evaluation metric, denoted as G-F1.

5.2.2 Reading Order

The ARD score (Wang et al., 2021) is proposed to evaluate the difference between reordered sequences. We extend the ARD score into a relative version to stabilize fluctuations caused by different paragraph splitting, denoted as RARD. The computation of RARD is detailed in section A.1.1.

5.2.3 Document Structure

We employ F1 of relation triplets (R-F1) to measure the correctness of document structure (Rausch et al., 2023). The structure relation is reconstructed from the depth of layout elements (Li et al., 2024). Details are provided in section A.1.2.

5.3 Comparison with LLM-based Methods

5.3.1 Open-source Baselines with SFT

When compared to open-source MLLMs with SFT, we evaluate several state-of-the-art models with priors about document parsing tasks, including GOT (Wei et al., 2024), InternVL3 (Zhu et al., 2025a), and Qwen2.5-VL (Bai et al., 2025). These models use different annotation schemes during pretraining: GOT and InternVL employ markdown output, while Qwen2.5-VL utilizes the QwenVL-HTML format to simultaneously obtain document parsing and grounding results. Since InternVL also supports grounding of the parsed content, we prepared document parsing data that includes grounding results following the format of InternVL, denoted as InternVL-grounding. All the models undergo SFT on the same corpus for one epoch.

The comparison results are detailed in table 1. The grounding results obtained by the IDP method significantly surpass those of other MLLMs. The

Method	HRDoc-Simple			HRDoc-Hard			DocHieNet		
	G-F1	ARD	R-F1	G-F1	ARD	R-F1	G-F1	ARD	R-F1
DocXChain	97.57	1.17	91.39	93.75	1.25	86.54	88.62	2.46	69.85
MinerU	98.37	0.83	-	96.39	1.08	-	85.82	2.83	-
Tencent	98.85	1.02	94.96	95.13	1.13	86.81	90.82	2.16	72.54
Mathpix	-	0.77	94.55	-	0.98	85.16	-	1.62	76.33
Ours	98.64	0.24	97.45	96.11	0.70	90.32	93.75	1.01	79.45

Table 2: Summary of performance of IDP model on different datasets compared to pipline-based document parsing methods.

Encoder	G-F1	ARD
DocXChain	66.23	5.25
Qwen-VL-Max	74.63	3.06
IDP	86.36	1.95

Table 3: Evaluation of zero-shot performance on the OmniDocBench dataset.

comparison between the IDP method and Qwen2.5-VL highlights the benefits of the decoupling paradigm and the training strategy of IDP. Additionally, markdown-based InternVL outperforms the grounding-based InternVL, despite markdown representation omitting multi-modal information such as images. This phenomenon further validates the contradiction between content parsing and layout grounding. The IDP method decouples the two tasks, allowing the MLLM to focus on recognizing and organizing textual content while retrieving multi-modal information through grounding tasks. Consequently, it achieves superior results over other MLLMs on different datasets.

5.3.2 Closed-source Baselines

We conducted tests on leading closed-source models GPT-4o[1] and the closed-source model with specifically priors for document parsing tasks Qwen-VL-Max (Bai et al., 2025). During the evaluation, we observe that the granularity of the paragraphs generated by the closed-source models may deviate from that of the annotations, making direct evaluation imprecise. Therefore, we combine overly segmented layout elements using rules following Ouyang et al. (2024). The post-processed evaluation results are presented in table 1.

It can be observed that the performance of the Qwen-VL-Max is inferior to that of smaller Qwen2.5-VL-7B which have undergone SFT, particularly in the grounding task, despite its parameter scale being much larger at 72B. The performance decline is partly due to severe shifted bounding boxes during grounding and approximately 5% failures at instructions following. The GPT-4o performs better, especially on datasets composed of scientific papers. However, there is a noticeable decline on the DocHieNet dataset, which consists of multi-type documents, particularly in the evaluation of hierarchical relationships.

5.4 Comparison with Pipeline Systems

In this section, we compare the IDP model with pipeline-based methods, DocXChain (Yao, 2023) and MinerU (Wang et al., 2024a), also with commercial APIs provided by Tencent[2] and Mathpix[3]. In addition, we augmented the DocXChain pipeline with a module for predicting document hierarchy (Ma et al., 2023), which was trained using DocHieNet and HRDoc datasets. The aforementioned matching strategy is also employed for evaluation.

It can be observed that the pipeline based on task-specific models performs well in layout element analysis, but shows no advantage in reading order and document structure results compared to the IDP approach due to error propagation. In fact, recent focus in the document parsing has been primarily on layout analysis (Chen et al., 2025), with only a few studies addressing order prediction and document structure parsing. The IDP approach constructs an effective end-to-end strategy to facilitate the progress of this part of research.

[1] https://platform.openai.com/

[2] https://cloud.tencent.com/document/product/1759/107506

[3] https://mathpix.com/

	k	G-F1	ARD	R-F1
IDP	4	93.75	1.01	79.45
w/o decouple	4	83.26	1.17	77.80
w/o stage3	4	87.82	1.01	79.14
IDP	3	91.59	1.02	79.44
IDP	1	79.73	1.01	79.47

Table 4: The ablation studies about key designs of the IDP model on the DocHieNet dataset.

5.5 Experiment on Zero-shot Benchmarks

We tested the zero-shot capability of different methods on a comprehensive multi-type document parsing benchmark, OmniDocBench. As shown in table 3, the MLLM-based models, such as Qwen and IDP, achieve better performance. Whereas the DocXChain method degrades in this zero-shot scenario, since the out-of-domain document types, such as newspapers and sides. Thanks to a specifically designed model and training stages, IDP outperforms the powerful Qwen-VL-Max in terms of zero-shot performance in the document parsing scenario.

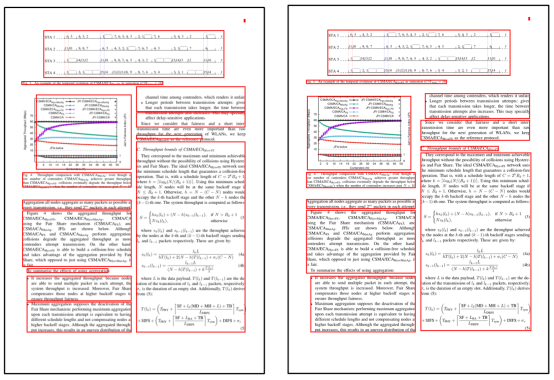
5.6 Ablation Study

In this section, we analyze the key designs in IDP through ablation experiments. We first attempt a model without the decoupled decoding process. As observed, the metrics for both tasks show significant declines compared to the decoupled format, validating the effectiveness of IDP's core concept. We also conduct control experiments without stage 3 training. It was evident that the grounding ability decreases. The transformer-based architectures impose high demands on convergence (Li et al., 2022), so the stage 3 is crucial for enhancing IDP's grounding capability.

Furthermore, we performed ablation experiments on the number of query tokens k, as shown in table 4. Insufficient k limits the representation capabilities of layout queries, affecting the accuracy of DLGM. As the number of query tokens increases, the marginal benefit decreases.

5.7 Qualitative Comparison of Layout Grounding Results

In fig. 4, the layout grounding inference results of the SFT-enhanced Qwen2.5-VL-7B and the IDP model are presented. Although the quantity of layout elements generated by Qwen-VL is mostly correct, there is a drifting phenomenon in the grounding results. These shifted boxes can significantly impact the parsing of multi-modal information, such as images, in the document. In contrast, the grounding results produced by the IDP method exhibit a high degree of alignment, addressing the issues observed with the sequence decoder. The appendix provides more qualitative results for a comprehensive comparison.



(a) QwenVL Result (b) IDP Result

Figure 4: Comparison of layout grounding results generated by spatial and sequential decoders.

	G-F1	ARD	R-F1
DocXChain-S	-	-	92.10
DocXChain	93.75	1.25	86.54
IDP	96.11	0.70	90.32

Table 5: The comparison among task-specific model (DocXChain-S), pipeline model (DocXChain) and IDP model on HRDH dataset.

5.8 Further Discussion on Pipeline Methods

In this section, we further compare the pipeline method DocXChain with the IDP model on the HRDH dataset in table 5. Specifically, by using the ground truth layout grounding information as input, we evaluate the model performance on the hierarchy parsing subtask, denoted as 'DocXChain-S'. The comparison with DocXChain with the whole pipeline quantifies the error propagation. It can be seen that the performance of DocXChain-S is better than the end-to-end results of IDP; however, the accumulated errors lead to a significant performance drop of DocXChain.

6 Conclusion

In this paper, we introduce the Intelligent Document Parsing (IDP) framework, an end-to-end document parsing framework leveraging the vision-language priors of MLLMs, equiped with elaborately designed IDP representation and fully activates the MLLMs ability on content parsing while generates the layout grounding results in a decoupled module. Experimental results demonstrate that IDP method effectively boosts the MLLM-based methods, and illustrate advantage to the pipeline methods in terms of end-to-end and zero-shot performance, providing a significant advancement in MLLM-based document parsing.

Limitations

Although IDP provides a powerful baseline for MLLM-based document parsing models, it does not consider the inter-page relationships in multi-page documents. In fact, information such as document layout style and hierarchical structure is related across pages. Moreover, in real-world scenarios, document parsing tasks typically involve multi-page document inputs. Expanding the IDP model to support joint inference across multiple pages is a potential area for future research.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*.

Yufan Chen, Ruiping Liu, Junwei Zheng, Di Wen, Kunyu Peng, Jiaming Zhang, and Rainer Stiefelhagen. 2025. Graph-based document structure analysis. *arXiv preprint arXiv:2502.02501*.

Hiuyi Cheng, Peiyu Zhang, Sihang Wu, Jiaxin Zhang, Qi Zhu, Zecheng Xie, Jing Li, Kai Ding, and Lianwen Jin. 2023. M6doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15138–15147.

Chris Clifton and Hector Garcia-Molina. 2000. The design of a document database. *Proceedings of the ACM conference on Document processing systems*.

Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. 2021. Document ai: Benchmarks, models and applications. *ArXiv*, abs/2111.08609.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Hao Feng, Shu Wei, Xiang Fei, Wei Shi, Yingdong Han, Lei Liao, Jinghui Lu, Binghong Wu, Qi Liu, Chunhui Lin, and 1 others. 2025. Dolphin: Document image parsing via heterogeneous anchor prompting. *arXiv preprint arXiv:2505.14059*.

Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. 2024. mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. *arXiv preprint arXiv:2409.03420*.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE.

Qing Jiang, Gen Luo, Yuqin Yang, Yuda Xiong, Yihao Chen, Zhaoyang Zeng, Tianhe Ren, and Lei Zhang. 2024. Chatrex: Taming multimodal llm for joint perception and understanding. *arXiv preprint arXiv:2411.18363*.

Stephen B. Johnson, David A. Campbell, M. Krauthammer, P. Karina Tulipano, Eneida A. Mendonça, Carol Friedman, and George Hripcsak. 2003. A native xml database design for clinical document research. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, page 883.

Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. 2022. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627.

Zhang Li, Yuliang Liu, Qiang Liu, Zhiyin Ma, Ziyang Zhang, Shuo Zhang, Zidun Guo, Jiarui Zhang, Xinyu Wang, and Xiang Bai. 2025. Monkeyocr: Document parsing with a structure-recognition-relation triplet paradigm. *arXiv preprint arXiv:2506.05218*.

Zichao Li, Shaojie He, Meng Liao, Xuanang Chen, Yaojie Lu, Hongyu Lin, Yanxiong Lu, Xianpei Han, and Le Sun. 2024. Seg2act: Global context-aware action generation for document logical structuring. *arXiv preprint arXiv:2410.06802*.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, and 1 others. 2024. Grounding dino:

Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Chuwei Luo, Yufan Shen, Zhaoqing Zhu, Qi Zheng, Zhi Yu, and Cong Yao. 2024. Layoutllm: Layout instruction tuning with large language models for document understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15630–15640.

Jiefeng Ma, Jun Du, Pengfei Hu, Zhenrong Zhang, Jianshu Zhang, Huihui Zhu, and Cong Liu. 2023. Hrdoc: dataset and baseline method toward hierarchical reconstruction of document structures. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Ye Mo, Zirui Shao, Kai Ye, Xianwei Mao, Bo Zhang, Hangdi Xing, Peng Ye, Gang Huang, Kehan Chen, Zhou Huan, and 1 others. 2025. Doc-cob: Enhancing multi-modal document understanding with visual chain-of-boxes reasoning. *arXiv preprint arXiv:2505.18603*.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, and 1 others. 2024. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. *arXiv preprint arXiv:2412.07626*.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed Samy Nassar, and Peter W. J. Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout segmentation. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. 2024. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13009–13018.

Johannes Rausch, Gentiana Rashiti, Maxim Gusev, Ce Zhang, and Stefan Feuerriegel. 2023. Dsg: An end-to-end document structure generator. *ArXiv*, abs/2310.09118.

Jon Saad-Falcon, Joe Barrow, Alexa F. Siu, Ani Nenkova, Ryan Rossi, and Franck Dernoncourt. 2023. Pdftriage: Question answering over long, structured documents. *ArXiv*, abs/2309.08872.

Ritesh Sarkhel and Arnab Nandi. 2019. Visual segmentation for information extraction from heterogeneous visually rich documents. In *Proceedings of the 2019 international conference on management of data*, pages 247–262.

Ritesh Sarkhel and Arnab Nandi. 2021. Improving information extraction from visually rich documents using visual span representations. *Proceedings of the VLDB Endowment*, 14(5).

Zirui Shao, Feiyu Gao, Zhaoqing Zhu, Chuwei Luo, Hangdi Xing, Zhi Yu, Qi Zheng, Ming Yan, and Jiajun Bu. 2024. Is cognition consistent with perception? assessing and mitigating multimodal knowledge conflicts in document understanding. *arXiv preprint arXiv:2411.07722*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, and 1 others. 2024a. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*.

Jiawei Wang, Kai Hu, and Qiang Huo. 2025. Unihdsa: A unified relation prediction approach for hierarchical document structure analysis. *Pattern Recognition*, page 111617.

Jiawei Wang, Kai Hu, Zhuoyao Zhong, Lei Sun, and Qiang Huo. 2024b. Detect-order-construct: A tree construction based approach for hierarchical document structure analysis. *ArXiv*, abs/2401.11874.

Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. 2021. Layoutreader: Pre-training of text and layout for reading order detection. *arXiv preprint arXiv:2108.11591*.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, and 1 others. 2024. General ocr theory: Towards ocr-2.0 via a unified end-to-end model.

Ross Wilkinson. 1994. Effective retrieval of structured documents. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jiannan Wu, Muyan Zhong, Sen Xing, Zeqiang Lai, Zhaoyang Liu, Zhe Chen, Wenhai Wang, Xizhou Zhu, Lewei Lu, Tong Lu, and 1 others. 2024. Visionllm v2: An end-to-end generalist multimodal large language model for hundreds of vision-language tasks. *Advances in Neural Information Processing Systems*, 37:69925–69975.

Hangdi Xing, Changxu Cheng, Feiyu Gao, Zirui Shao, Zhi Yu, Jiajun Bu, Qi Zheng, and Cong Yao. 2024. Dochienet: A large and diverse dataset for document hierarchy parsing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1129–1142.

Hangdi Xing, Feiyu Gao, Rujiao Long, Jiajun Bu, Qi Zheng, Liangcheng Li, Cong Yao, and Zhi Yu. 2023. Lore: Logical location regression network for table structure recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2992–3000.

Cong Yao. 2023. Docxchain: A powerful open-source toolchain for document parsing and beyond. *arXiv preprint arXiv:2310.12430*.

Qintong Zhang, Victor Shea-Jay Huang, Bin Wang, Junyuan Zhang, Zhengren Wang, Hao Liang, Shawn Wang, Matthieu Lin, Conghui He, and Wentao Zhang. 2024. Document parsing unveiled: Techniques, challenges, and prospects for structured information extraction. *arXiv preprint arXiv:2410.21169*.

Xu Zhong, Jianbin Tang, and Antonio Jimeno-Yepes. 2019. Publaynet: Largest dataset ever for document layout analysis. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, and 1 others. 2025a. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

Jinguo Zhu, Xizhou Zhu, Wenhai Wang, Xiaohua Wang, Hongsheng Li, Xiaogang Wang, and Jifeng Dai. 2022. Uni-perceiver-moe: Learning sparse generalist models with conditional moes. *Advances in Neural Information Processing Systems*, 35:2664–2678.

Zhaoqing Zhu, Chuwei Luo, Zirui Shao, Feiyu Gao, Hangdi Xing, Qi Zheng, and Ji Zhang. 2025b. A simple yet effective layout token in large language models for document understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14472–14482.

A Appendix

A.1 Evaluation Metric

A.1.1 Reading Order

The ARD score (Wang et al., 2021) is proposed to evaluate the difference between reordered sequences. We extend the ARD score into a relative version to stabilize fluctuations caused by different paragraph splitting, denoted as RARD.

Given a sequence $A = [e_1, e_2, \ldots, e_n]$ and the prediced sequence $B = [e_{i_1}, e_{i_2}, \ldots, e_{i_m}]$, where $\{i_1, i_2, \ldots, i_m\} \subseteq \{1, 2, \ldots, n\}$, the ARD score is calculated as follows:

$$s(e_k, B) = |\frac{k}{n} - \frac{I(e_k, B)}{m}|$$

$$\text{RARD}(A, B) = \frac{1}{n} \sum_{e_k \in A} s(e_k, B)$$

where e_k is the k-th element in sequence A; $I(e_k; B)$ is the index of e_k in sequence B; n is the length of sequence A.

A.1.2 Document Structure

We employ F1 of relation triplets (R-F1) to measure the correctness of document structure (Rausch et al., 2023). The structure relation is reconstructed from the depth of layout elements (Li et al., 2024).

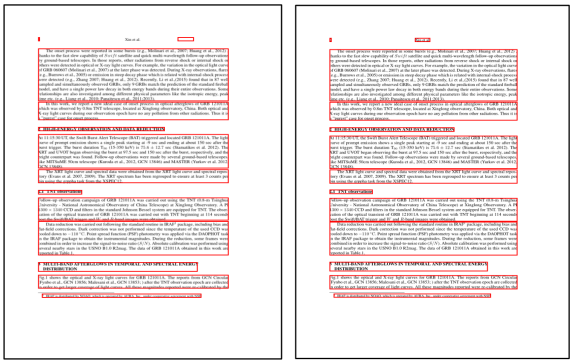
Suppose $R_{gt} = \{(E_{parent}, E_{child}, r_{gt})\}$ and $R_{pred} = \{(\hat{E}_{parent}, \hat{E}_{child}, \hat{r}_{pred})\}$, then the F1-score is computed from the precision p_{score} and recall r_{score} as following:

$$p_{score} = \frac{|R_{gt} \cap R_{pred}|}{|R_{pred}|}, r_{score} = \frac{|R_{gt} \cap R_{pred}|}{|R_{gt}|}$$

A.2 Qualitative Results

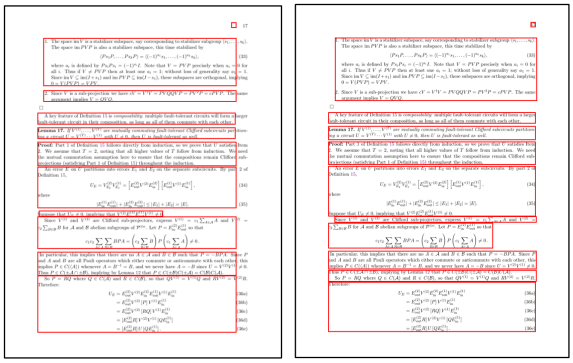
In this section, we present more qualitative comparison of grounding results between MLLM-based method, Qwen-VL with SFT and the IDP method. As depicted in fig. 5, the bounding boxes generated by QwenVL exhibit hallucinations. Specifically, QwenVL perceives and parses the header information on the page, but the predicted header box is positioned far from the actual header on this page, likely due to Qwen-VL hallucinating based on the typical location of headers in the data.

In the sample shown in fig. 6, the boundaries on the left, top, and bottom sides are reasonable, yet the right boundary shows an overall displacement. The sequential inference mechanism of MLLM may cause errors that influence the subsequent grounding results of other layout elements.



(a) QwenVL Result (b) IDP Result

Figure 5: Visualization of the hallucinations grounding results generated by MLLM compared to IDP.



(a) QwenVL Result (b) IDP Result

Figure 6: Visualization of the shifted grounding results generated by MLLM compared to IDP.