
ParseFormer: A Multi-Task Framework for Chinese Scientific PDF Parsing

Guangyin Zhang^{1,2}, Liping Gu^{1,2}, Meng Wang^{1,2}, Mengting Zhang^{1,2}, Yang Zhao^{1,2}, Maodi Hu^{1,2},
Zhixiong Zhang^{1,2}

1.National Science Library, Chinese Academy of Science, Beijing 100190, China

2.Department of information Resources Management, School of Economics and Management,
University of Chinese Academy of Sciences, Beijing 100190, China

Keywords: Scientific PDF Parsing; Reading Order Detection; Text Function Identification; Multi-task Learning; Citation Extraction; Paragraph Boundary Identification

Abstract

Accurate parsing of scientific PDFs is essential for effective knowledge dissemination and utilization, supporting applications such as citation analysis and retrieval-augmented systems. However, existing solutions focus predominantly on English documents and struggle with Chinese academic journal PDFs due to their unique features, such as bilingual titles/abstracts and page concatenation. There is also a lack of publicly available datasets for full-text parsing of Chinese academic journals. To bridge this gap, we first constructed a specialized Chinese scientific PDF dataset comprising 2613 annotated papers, across 14 disciplines, featuring 16 fine-grained text function labels tailored to Chinese scientific journal characteristics. Leveraging this dataset, we propose ParseFormer, a novel multi-task learning model based on the LayoutLMv3 architecture, which simultaneously performs four key tasks: reading order detection (ROD), text function identification (TFI), paragraph boundary identification (PBI), and citation extraction (CE). To predict the reading order more efficiently, we propose a regression head trained with the List-MLE function to replace the common pairwise ordering method. Experiments demonstrate the effectiveness of the proposed approach, achieving F1-scores above 0.92 across all 16 main TFI labels, 0.98 for PBI, 0.82 for CE, and an ARD of 0.039 for ROD. This framework achieves a good balance between efficiency and accuracy, facilitating conversion of PDFs into structured JSON or Markdown formats. We hope this work can serve as a useful resource for Chinese scientific PDF mining.

[†]Corresponding author: Liping Gu (Email: gulp@mail.las.ac.cn; ORCID:0000-0002-2284-3856).

1. Introduction

Scientific documents serve as a cornerstone of academic knowledge dissemination, embodying rich theories, methodologies, and research findings in their full text, and are most commonly distributed in Portable Document Format (PDF). The increasing sophistication of modern analytical applications necessitates the processing of entire scientific documents rather than relying solely on metadata or abstracts. Consequently, the ability to accurately parse and structure complete scientific PDFs has become a critical prerequisite for advanced literature mining. Such capability enables various downstream applications, including extended citation analysis [1], [2] which requires both citation contexts and reference lists; the construction of scientific domain knowledge bases [3], [4] where abstract-only analysis is insufficient; and the development of retrieval-augmented question answering systems [5], which demand sophisticated document parsing and chunking strategies.

Despite rapid progress in general PDF parsing, effective solutions for Chinese scientific journals remain limited. Existing domain-specific parsers, such as GROBID [6], are predominantly trained on English documents. Some of its input features, such as reliance on capitalization of the first letter, are only suitable for Latin-based languages and not for Chinese. General-purpose PDF parsers, while powerful in layout segmentation and table recognition, produce outputs that are too coarse-grained for the needs of scientific content analysis. Moreover, Chinese academic journal PDF documents often contain element types that differ from those found in English documents. For example, in addition to the Chinese versions of the title and abstract, they also include English versions of the title and abstract. Furthermore, Chinese journal PDF documents frequently exhibit page concatenation, where content from two different articles is placed on the same page. Therefore, developing a dedicated parser for Chinese academic PDFs is crucial for building comprehensive knowledge bases and advancing research based on Chinese scientific literature. In addition, most existing publicly available Chinese scientific PDF datasets contain only metadata, such as titles and abstracts, and there is a notable lack of interdisciplinary datasets with full text.

To address these gaps, we first construct a dedicated dataset of Chinese scientific journal PDFs collected from open-access sources across 14 disciplines. Each article is annotated with fine-grained structural features—including 16 text function labels, paragraph boundaries, citation spans, and page concatenation markers—reflecting the distinctive layout and semantics of Chinese journals. Building upon this dataset, we propose ParseFormer, a unified multi-task learning model that jointly leverages semantic and positional features to perform four key tasks: reading order detection (ROD), text function identification (TFI), paragraph boundary identification (PBI), and citation extraction (CE). On top of ParseFormer, we further design a full-text parsing pipeline that integrates layout analysis, OCR, and structured conversion modules. This pipeline takes advantage of ParseFormer’s predictions to transform Chinese journal PDFs into reliable machine-readable formats such as JSON and Markdown, thereby directly addressing the issues of bilingual structures, page concatenation, and fine-grained content extraction highlighted above. The resulting system provides a solid foundation for downstream applications including citation analysis, knowledge base construction, and retrieval-augmented systems.

The contributions of this work are twofold:

- Chinese scientific journal PDF dataset: We construct the first interdisciplinary dataset of Chinese scientific journal PDFs, comprising 2613 articles and 12793 pages. The dataset is annotated with 16 fine-grained text function labels, paragraph boundaries, citation spans, and page-concatenation tags, tailored to Chinese scientific journal features like bilingual titles/abstracts and inter-article merging. This resource fills a critical gap and provides a foundation for research on automated parsing and mining of Chinese academic literature.
- ParseFormer, a unified multi-task parsing model: We propose ParseFormer, a deep learning model that integrates the four parsing tasks—ROD, TFI, PBI and CE—within a shared encoder. In particular, ParseFormer introduces a listwise ranking approach with List-MLE loss for efficient reading order prediction. Experiments demonstrate that this unified design achieves high accuracy while maintaining efficiency, enabling robust conversion of Chinese scientific PDFs into structured formats.

2. Related Work

In this section, we first give an overview of existing scientific PDF parsing methods and their limitations for Chinese academic journals, and then discuss related studies on the four subtasks addressed in our framework: reading order detection, text function identification, paragraph boundary identification, and citation extraction.

2.1 General Overview of Scientific PDF Parsing

PDF parsing refers to the task of transforming visually formatted PDF files into machine-readable structured data to support downstream applications such as information retrieval, knowledge base construction, and question answering. For general Chinese documents such as news articles, contracts, or reports, parsing mainly involves recovering high-level layouts, extracting textual blocks, and reconstructing tables or forms, which typically involves less structural complexity compared to scientific articles. In contrast, Chinese scientific journal articles contain more structural elements, such as the bilingual titles and abstracts, author–affiliation mappings, keywords, funding statements, figure and table captions, and references. In addition, the cross-article concatenation is another special phenomenon in Chinese scientific journal documents. In general Chinese documents, page concatenation usually occurs by joining the ending of one article with the beginning of the next, whereas in scientific documents it may involve placing the endings of two different articles together on the same PDF page. Table 1 shows the qualitative comparison between general Chinese documents and scientific documents. These distinctive components make academic literature a fundamentally different research object from ordinary Chinese documents and require a different parsing approach.

Existing parsing methods can be broadly categorized according to their methodological design into pipeline approaches and end-to-end approaches. Pipeline methods decompose document parsing into a sequence of modules such as OCR, layout analysis, and post-processing rules. Examples include GROBID [6], PaddleOCR-ppstructure [7], Marker [8], and MinerU [9]. By contrast, end-to-end approaches aim to learn unified multimodal document representations and directly generate structured outputs, thereby reducing error propagation across modules. Representative works include GOT-OCR [10], Nougat [11], MonkeyOCR [12], SmolDocling [13], olmOCR [14] etc. Benefiting from the multi-modal document large language models, these models typically use an image encoder to encode document images, then use a text decoder to output the content of the entire document as a one-dimensional sequence with markdown format. Among the above two types of approaches, only GROBID is a domain-specific parser designed for scholarly documents, capable of accurately extracting bibliographic metadata and references from English corpora, but its reliance on language-specific cues makes it unsuitable for Chinese scientific documents. In contrast, other works are general-purpose PDF parsing frameworks applicable across multiple document types, while they perform well in layout segmentation and table or figure detection, their outputs are too coarse to capture the detailed structures of academic PDFs, leading to missing or insufficient annotations when applied to Chinese journals. As a result, both domain-specific and general-purpose approaches face limitations in handling the structural complexity of Chinese scientific papers, underscoring the necessity of a dedicated framework. In the following subsections, we review prior work across four subtasks—reading order detection, text function identification, paragraph boundary identification, and citation extraction—as they represent both the fundamental capabilities required for document parsing and a key application-driven objective.

Table 1. Qualitative comparison between general Chinese documents and Chinese scientific journal documents, highlighting structural features that make the latter a distinct research object for PDF parsing.

Feature / Element	General Chinese Documents	Chinese Scientific Journal Articles
Bilingual titles & abstracts	✗ Absent	✓ Mostly
Author–affiliation mapping	✗ Rare	✓ Mostly
Bilingual Keywords section	✗ Rare	✓ Mostly
Funding / project statements	✗ Rare	✓ Mostly (Often required by journals)
In-text citations & reference list	✗ Rare / absent	✓ Mostly (critical scientific component)
Page concatenation across articles	✓ (Continuous: article endings and beginnings stitched together)	✓ (Cross-page: endings of different articles stitched together)
Complex multi-column layout	✓ Mostly (one/two/three column)	✓ Mostly (one/two column)
Author introduction at end page	✗ Rare	✓ Often

2.2 Reading Order Detection

Reading order detection (ROD) aims to determine the correct sequence in which text elements (such as paragraphs, captions, or columns) should be read in a document or image. Its goal is to reconstruct the natural flow of reading, enabling machines to interpret multi-column layouts, complex page structures, or mixed text and graphics accurately. Early studies mainly relied on heuristic rule-based approaches [15]. With the development of deep learning, methods have evolved into two branches: non-generative and generative. The non-generative approach usually encodes text input through a deep network encoder and then predict the reading order through pairwise order relationship prediction [16], [17], [18]. These models typically take textual and positional features as input. Some variants [19] further incorporate visual features extracted from text images. They ultimately construct a pairwise order relationship matrix and decode the global reading order based on various kinds of devised rules applied to this matrix.

The second type of approach predicts the reading order of document pages through a generative way which predicts the sequence one by one. The earliest work can trace to [18] which represents text blocks as nodes in a graph, encodes both visual and geometric layout features using a Graph Convolutional Network (GCN) with attention, and then applies a pointer-based decoder with Sinkhorn normalization to generate the final reading sequence. LayoutReader [20] adopts an encoder-decoder architecture. It inputs the characters and two-dimensional position information of the text, and the decoder outputs the correct reading order of the text. Nougat [11] uses the SWIN-transformer [21] architecture as the image encoder and Bart as the decoder to generate document reading text. More recent works correspond the end to end approaches described in Section 2.1 [10], [13], [14]. Compared to the previous method [20], these approaches usually take document OCR parsing as a basic training task, eliminating the need for using OCR to extract text coordinates as a preliminary step, instead generating text directly from images. Generally, these multi-modal large models require large training sets, millions of pages or more, and longer training times compared to pure text models. Generative models may also produce hallucinated outputs, either omitting certain characters or generating characters that are not present in the original page.

2.3 Text Function Identification

Text function identification (TFI) is also critical to producing structured outputs from scientific PDFs.

TFI assigns semantic roles to text blocks, enabling fine-grained structuring of document content. A representative system is GROBID [6], which uses cascaded sequence labeling with a large set of function labels. However, GROBID is trained exclusively on English corpora and relies on language-specific cues such as capitalization, which are ineffective for Chinese and their performance degrades on scanned PDFs. General PDF parsing methods usually give the text function labels base on their layout analysis results [8], [9]. For example, DocLayout-YOLO [22] leverages the diverse synthetic dataset DocSynth-300K and a global-to-local controllable receptive module (GL-CRM) to significantly improve document layout analysis accuracy while maintaining high speed. These kinds of methods detect the text function labels mainly based on the visual features and their label granularity is insufficient for detailed academic parsing. In contrast, our work supports 16 fine-grained labels, explicitly tailored to Chinese journal layouts, by leveraging both semantic and positional features within a unified encoder.

2.4 Paragraph Boundary Identification

Paragraph Boundary Identification (PBI) determines whether a text line begins a new paragraph, a seemingly simple task that underpins the preservation of logical structure. Existing solutions often rely on the paragraph boundary detected by the layout analysis model and handcrafted rules derived from indentation or spacing features [8], [9]. This method is not robust enough especially when dealing with the reference list blocks. We instead treat PBI as a supervised binary classification problem at the text-line level, allowing the model to capture contextual cues that rule-based methods overlook.

2.5 Citation Extraction

Citation Extraction (CE) identifies in-text citation spans, which is essential for building citation networks and linking references. This task has been extensively studied for many years. Traditional approaches use Hidden Markov Models [23], Conditional Random Fields [6], [24] to match citation patterns. More recently, LLM-based methods can generalize better but tend to over-predict by labeling non-citation mentions. In Chinese scientific texts, the diversity of citation styles and OCR errors—especially in superscript digits—further complicate extraction. In this work, we formulate CE as a sequence-labeling problem using the BIO tagging scheme, which enables precise localization of both numeric and author-year citations, while also leveraging the shared contextual representations within our multi-task framework.

3. Chinese scientific PDF dataset

3.1 Dataset Construction

To train and evaluate our unified parsing model, we constructed a dedicated dataset of Chinese scientific journal PDFs. The dataset annotation follows a semi-automatic pipeline (Figure 1), integrating existing PDF parsing tools, multimodal large language models (MLLMs), general-purpose LLMs, and human verification. We collected 2613 open-access journal articles from [25], spanning 14 disciplines, including physics, chemistry, biology, medicine, economics, management, geography, and agronomy etc. Figure 2 illustrates the distribution of papers across disciplines. To ensure a balanced representation, the proportion of each discipline was controlled within the range of 5%–8%. The pipeline consists of four steps, each focusing on a specific type of information.

Step 1: Reading Order Annotation.

In the first step, the goal was to determine the sequential order in which text lines should be read on each page, which is essential for reconstructing coherent content flow in multi-column or complex layouts. Candidate reading sequences were first generated using general-purpose PDF parsers such as MinerU [9] and PP-Structure [7]. These tools combine OCR with layout analysis to generate line-level reading sequences. The initial outputs often contained errors (e.g., incorrect column ordering, misplaced captions) and thus underwent manual verification and correction.

Step 2: Text Function and Paragraph Boundary Annotation

This step aimed to label the functional role of each text line (e.g., article title, abstract, keywords, body text, references) and identify whether the line marked the beginning of a new paragraph, so that the fine-grained document structure could be recovered. MinerU only provides coarse-grained text labels,

which are insufficient for academic parsing. We therefore expanded the schema to 16 fine-grained text function categories (see Table 2), reflecting the unique structure of Chinese scientific journals. The detailed explanation of each label can be seen in Appendix A. To accelerate annotation, we employed multimodal LLMs (Qwen-VL, GPT-4o), which were prompted to assign both text function labels and paragraph boundaries at the line level. All outputs were further manually verified to ensure consistency and correctness. The specific annotation prompt can be found in Appendix B.

Step 3: Citation Annotation

Here the objective was to locate explicit in-text citations in both body text and figure/table captions, which are necessary for downstream bibliometric analysis and citation network construction. We used LLMs (DeepSeek, GPT-4o) to identify citation spans in both body text and captions. Two common formats were annotated: Numeric citations (e.g., “[1]”, “[2–3]”) and Author–year citations (e.g., “(Smith, 2020)”). Other highly irregular citation styles (frequent in philosophy and literature field) were excluded and left for future work. Annotation was conducted at the character level using the BIO tagging scheme, enabling precise localization of citation spans.

Step 4: Page Concatenation Annotation

The final step focused on identifying cases where a single PDF page contained text from multiple articles, a phenomenon especially common in Chinese journals on the first or last page. Such cases are usually marked by phrases like “上接第XXX页” or “下接第XXX页”. Figure 3 shows two typical examples of the page concatenation. For such pages, text lines were annotated with continuation markers at the line level to distinguish content belonging to different articles. Since existing multi-modal large language models still struggle to robustly detect such boundaries, this step was performed manually by human annotators.

The dataset was annotated at multiple levels of granularity: reading order, text function, paragraph boundaries, and page concatenation were labeled at the line level (based on OCR-detected text lines), while citation spans were annotated at the character level using the BIO tagging scheme. To further improve annotation quality, we also adopted an iterative correction strategy. After an initial round of model training, samples that yielded unusually high loss values or inconsistent predictions were automatically flagged as potentially mislabeled. These samples were then manually reviewed and corrected by human annotators, after which the revised dataset was used for retraining the model. This detection–correction–retraining cycle was repeated multiple times, progressively reducing annotation noise and improving both the consistency and reliability of the dataset.

During the manual annotation phase, the annotators were two second-year doctoral students in the field of library and information science, both of whom are co-authors of this paper. After jointly discussing and agreeing on detailed annotation guidelines with explicit definitions and examples, the two annotators achieved Cohen’s Kappa values above 0.90 over the trial annotation batches, indicating a very high level of consistency. Based on this result, they subsequently divided and collaboratively completed the remaining annotation work in separate batches.

3.2 Dataset Statistics

Table 2 reports some statistics about annotated text function labels. Compared to existing Chinese PDF datasets, ours provides comprehensive, fine-grained annotations tailored for academic journal parsing, covering 16 function labels, paragraph boundaries, citation spans, and page concatenation cases. Table 3 reports some general statistics of the constructed dataset. In total, the dataset contains 2,613 PDFs with 12,793 pages, of which 2,090 PDFs are used for training, 261 PDFs are reserved for validation, and 262 PDFs are allocated for testing.

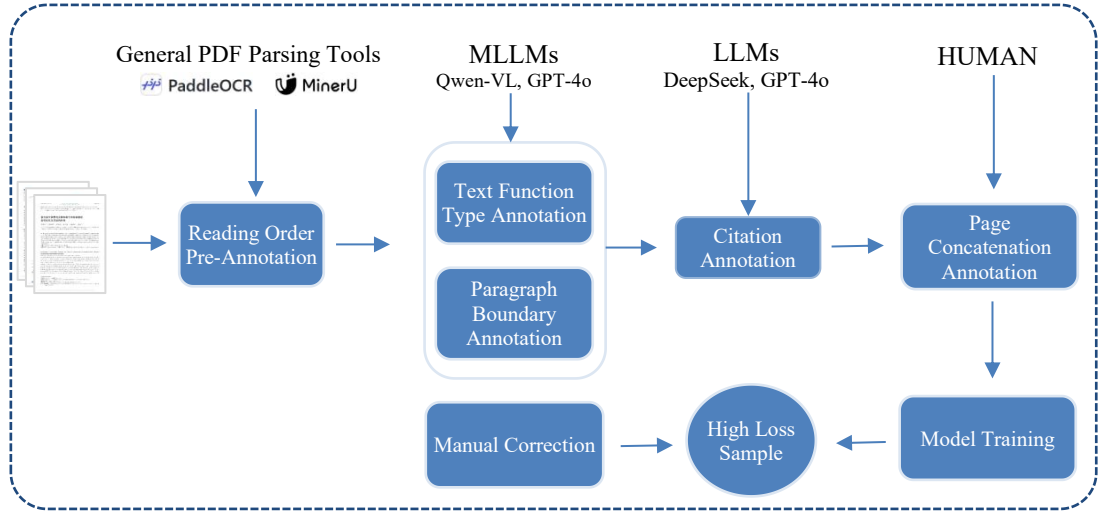


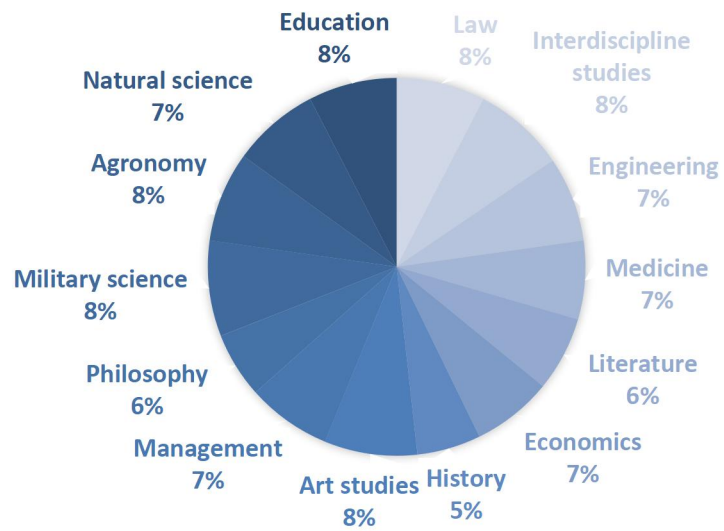
Figure 1. The semi-automatic annotation pipeline for constructing the Chinese scientific PDF dataset. The process integrates general PDF parsers, multi-modal LLMs, and human verification.

Table 2. Statistics of 16 types of annotated text function labels at the text line level

Type	Description	Train set	Val set	Test set
Table caption	Table caption (includes the table footnote)	2353	328	340
Figure caption	Figure caption	3623	399	375
Paragraph text	Body text	436159	52718	55651
Chinese Article Title	Article title with Chinese language	2683	327	355
English Article Title	Article title with English language	1229	158	171
Section Title	Section title	18303	2266	2188
Chinese Abstract	Abstract with Chinese language	11288	1440	1410
English Abstract	Abstract with English language	7937	1054	1071
Chinese author name	Author name with Chinese language	2253	268	275
English author name	Author name with English language	696	90	100
Chinese Keywords	Keywords with Chinese language	2066	253	258
English Keywords	Keywords with English language	916	117	124
Chinese Institution	Institution with Chinese language	1881	239	242
English Institution	Institution with English language	850	135	134
Reference	Reference (the reference entry in page footnote is excluded)	42788	5509	5424
Other type	include content concatenated from other articles, header, footer	62526	8031	8106

Table 3. Statistics of the constructed PDF dataset

	Train set	Val set	Test set
Doc num	2090	261	262
Page num	10204	1277	1312
Table num	2224	290	257
Figure num	2510	336	264
Equation num	1344	213	188
Page with concatenation	875	74	76
Citation num	16882	1217	1430

**Figure 2.** Disciplines distribution in the constructed PDF dataset

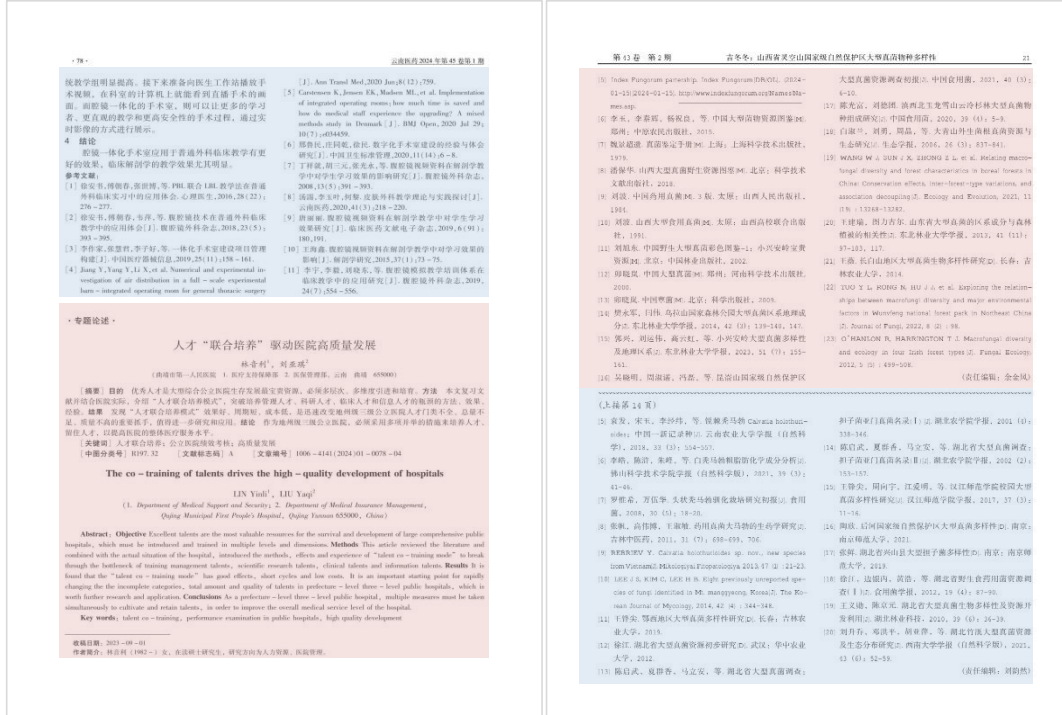


Figure 3. Example of page concatenation. The left page contains the beginning of one paper and the end of another paper. The right page contains the end of one paper and the end of another.

4 PDF-parsing using multi-task learning

4.1 Overview

To overcome the limitations of existing solutions and enable accurate extraction of Chinese scientific PDFs, we developed a new parsing pipeline. Figure 4 illustrates the main workflow of proposed full-text parsing pipeline. Firstly, a layout analysis model is applied to the PDF document image to identify three types of non-text elements: figures, tables, and interline formulas. These areas are then masked to prevent interference with subsequent steps. While some document layout analysis models can detect a wider range of textual elements (e.g., Titles, Captions), we found that their accuracy on text blocks does not meet our requirements when applied on our newly constructed PDF dataset. Therefore we only use the layout analysis model to detect these three types of elements. The second step is to apply OCR on masked PDF pages to detect text lines, recognize text content and obtain coordinates of text lines. The third step is to use a devised deep model to detect reading order, identify the text function of each line, determine paragraph boundaries, and extract citation spans. The fourth step is based on the model predictions to further convert the content into JSON and Markdown format. The models used in Steps 1 and 2 are based on publicly available pre-trained models. Our core contribution is the design and training of the main parsing model. We develop this model based on the LayoutLMv3 [26] architecture and design it to perform the following four tasks:

- **Reading Order Detection (ROD):** Predict the correct sequential order of text lines within each PDF page. Accurate reading order is fundamental for reconstructing coherent document flow, especially in multi-column layouts.
- **Text Function Identification (TFI):** Assign semantic roles (e.g., title, abstract, body text, caption) to each text block. This enables fine-grained structuring of the document and supports downstream tasks such as citation analysis and knowledge base construction.
- **Paragraph Boundary Identification (PBI):** Decide whether a given text line marks the start of a new

paragraph. Reliable paragraph segmentation preserves the logical structure of the text and cannot be achieved robustly by rule-based heuristics alone.

- Citation Extraction (CE): Detect citation spans in both body text and captions, covering numeric (e.g., “[1], [2–3]”) and author–year (e.g., “(Smith, 2020)”) formats. This task is critical for building citation networks and linking references.

These four tasks were chosen because they represent both the foundation and the practical extension of scientific PDF parsing. Among them, ROD, TFI, and PBI are fundamental capabilities that any complete parsing solution should directly or indirectly include, since they reconstruct the reading flow, assign functional roles to text, and preserve paragraph-level discourse structure. CE, on the other hand, is emphasized in our framework due to strong needs in bibliometric analysis and institutional project requirements. Notably, the design of ParseFormer is not limited to these four tasks; the unified encoder architecture allows new objectives—such as heading hierarchy recognition or reference entry parsing—to be seamlessly added in future work.

To better highlight the novelty of our approach, Figure 4 contrasts our pipeline with general PDF parsing methods. The main differences can be summarized in two aspects:

- Selective layout analysis: General PDF parsing methods typically apply the layout analysis model to detect all types of page elements. While our method only leverages layout analysis results for three element types—tables, figures, and interline equations. This selective use avoids the unreliable predictions when applying general-purpose layout detectors to our dataset;
- Unified encoder model: General PDF parsing methods usually do not employ a deep encoder for text labeling. In this paper, we introduce ParseFormer, a unified deep encoder that jointly processes all page elements (text lines and special tokens). ParseFormer not only supports the four parsing tasks—ROD, TFI, PBI, and CE—but also provides flexibility for future extensions to additional labeling tasks.

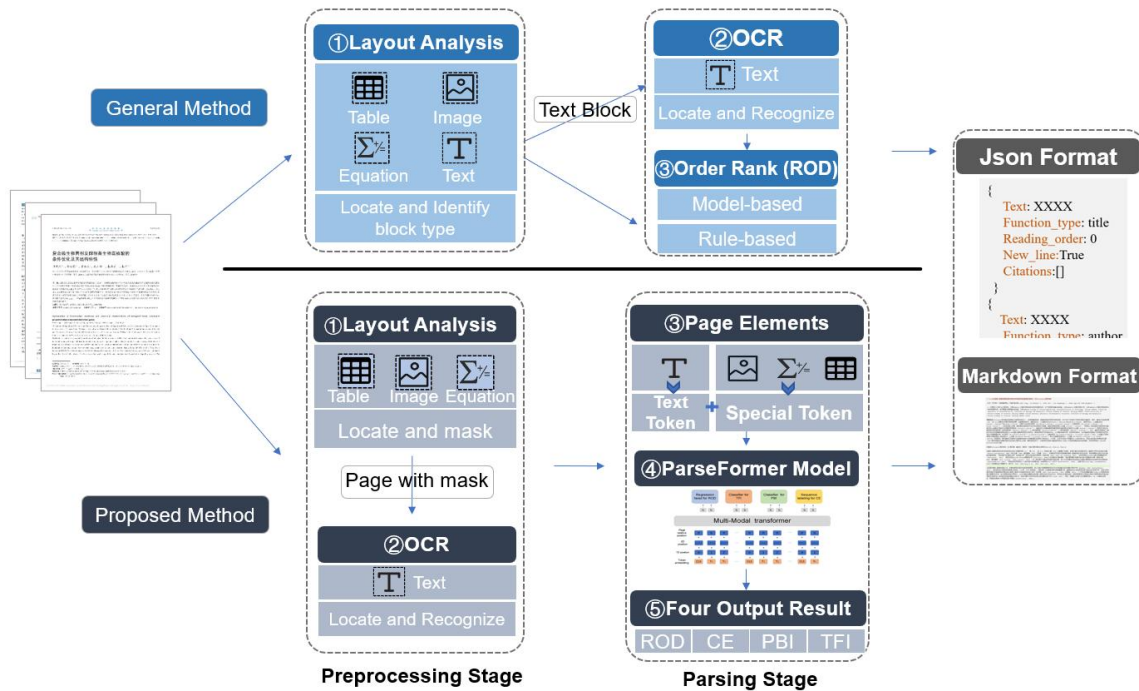


Figure 4. The comparison between proposed PDF parsing pipeline and general PDF parsing pipeline

4.2 ParseFormer Model

The overall architecture of the ParseFormer model is illustrated in Figure 5. Given a scientific PDF document, we first decompose it into a sequence of pages, denoted as $D = \{P_1, P_2, \dots\}$, where P_i denotes the i -th page. For each page, layout analysis is first applied to detect non-text elements, including figures, tables, and interline equations. The detected non-text elements on page P_i are denoted as $G_i = \{g_i^1, g_i^2, \dots\}$, where g_i^j can be a figure, table or interline-equation. $b_i^j = (x_{min}, y_{min}, x_{max}, y_{max})$ is the bounding box of element g_i^j . After masking these non-text regions, OCR is performed on the page to extract textual content. The OCR output on page P_i is represented as $T_i = \{t_i^{k1}, t_i^{k2}, \dots\}$, where t_i^{km} denotes the m -th token in k -th line of i -th page. $b_i^k = (x_{min}, y_{min}, x_{max}, y_{max})$ denotes the bounding box of the k -th line in i -th page.

To jointly model textual content, page-level context, and spatial layout information, the ParseFormer model extends the LayoutLMv3 [22] embedding layer to incorporate multiple positional and semantic signals. Specifically, for each element in set G_i and each text token in set T_i , the final embedding vector is computed as the sum of four types of embeddings: token embedding, 1D positional embedding, page position embedding, and 2D spatial embedding.

- Token embedding represents the token itself. Non-text elements, including figures, tables, and interline formulas, are all transformed to one special token to get their own embeddings. A [CLS] token is inserted at the beginning of each OCR text-line sequence;
- 1D positional embedding encodes the token’s sequential position within its text line. Indexing starts from 0 for every line. The [CLS] token is always assigned position 0, while text tokens t_i^{km} in the line are indexed starting from 1. For non-text elements in set G_i , their 1D position is also fixed to 0.
- Page position embedding captures the relative position of the page in the document, computed as:

$$page_pos = \frac{page_index}{total_pages} \times R \quad (1)$$

where R is a fixed scaling range. This feature captures the observation that certain semantic elements (e.g., references, headers) tend to appear at specific page positions.

- 2D spatial embedding incorporates the bounding box coordinates and the corresponding height and width. Given a bounding box on a page of width W and height H , its raw coordinates $(x_{min}, y_{min}, x_{max}, y_{max})$ are first normalized to the page size and then mapped to a fixed discrete range $[0, S]$ as follows:

$$\tilde{x} = \left\lfloor \frac{x}{W} \cdot S \right\rfloor, \quad \tilde{y} = \left\lfloor \frac{y}{H} \cdot S \right\rfloor \quad (2)$$

Accordingly, the discretized width and height are defined as:

$$\tilde{w} = \tilde{x}_{max} - \tilde{x}_{min}, \quad \tilde{h} = \tilde{y}_{max} - \tilde{y}_{min} \quad (3)$$

The 2D spatial embedding is then computed as

$$SpatEmb_{b_{ik}} = concat(E_x(\tilde{x}_{min}), E_y(\tilde{y}_{min}), E_x(\tilde{x}_{max}), E_y(\tilde{y}_{max}), E_h(\tilde{h}), E_w(\tilde{w})) \quad (4)$$

Where E_x, E_y, E_h, E_w are separate embedding tables. Tokens belonging to the same text line share the same 2D spatial embedding.

Formally, for m -th text token in k -th line in i -th page, the input embedding is:

$$e_i^{km} = TokEmb(t_i^{km}) + PosEmb1D(m+1) + PageEmb(p_i) + SpatEmb(b_i^k) \quad (5)$$

where p_i is the normalized page position, and b_i^k denotes the bounding box of the token’s text line. All embedding components are summed element-wise.

For element g_i^j in set G_i , the input embedding is:

$$e_i^j = TokEmb(g_i^j) + PosEmb1D(0) + PageEmb(p_i) + SpatEmb(b_i^j) \quad (6)$$

where each type of g_i^j is transformed to its corresponding one special token to get its token embedding.

The encoder employs standard multi-head self-attention layers, as the self-attention mechanism inherently allows each element to incorporate information from all others. Each time, we input all the tokens in two consecutive PDF pages and the model make predictions for the latter page. This is because the model needs to check the content in two pages when making predictions on whether there is concatenated content. After getting the hidden layer representation, it will go through four different

processing heads for the four tasks.

The first head is a regression layer for ROD task. Unlike the commonly used pairwise approach, we propose a regression-based approach for ordering. This can improve inference speed by discarding pairwise relationship prediction and directly outputting ranking scores. For pairwise relationship prediction, the computational complexity is $n * (n - 1)/2$, while the method proposed in this paper has a computational complexity of n (n is the number of sequences). We believe that the attention mechanism in transformers is sufficient for the hidden layer representations of tokens to carry surrounding contextual positional information, and there is no need to predict a binary relationship matrix at the output. So we directly use a MLP as a regression head to predict the rank score of each element. The order unit is each text line and equation line detected by OCR. The input of this head is the hidden representation of the [CLS] token which is added in front of each line from the input. The output is a regression score for each [CLS] token. The higher regression score, the higher the ranking position. The List-MLE [27] loss function is applied to train this method. Eq.(7) gives the mathematic expression of the List-MLE function.

$$L(\theta) = -\log P(\sigma|\theta) = -\sum_{k=1}^n \log \left(\frac{\exp(f(x_{\sigma(k)}; \theta))}{\sum_{j=k}^n \exp(f(x_{\sigma(j)}; \theta))} \right) \quad (7)$$

where θ is the model's trainable parameters, $f(x_{\sigma(k)}; \theta)$ is the ranking score of text line k predicted by the regression head and $P(\sigma|\theta)$ serve as the probability of the ground truth rankings. The training aim is to increase the probability of the ground truth ranking. Besides, the order loss between floating elements (including table captions and figure captions) and fixed elements (including titles and paragraph text) is calculated separately, since we argue that these floating elements should not have a fixed order relationship with paragraph text. The order loss for other types of text lines, such as page headers, page footers, and concatenated content, is not calculated.

The second head is a text type classifier with 16 classes for TFI task. This classifier is implemented as a three-layer MLP followed by a softmax function. The input is the hidden representation of [CLS] token. It has a cross entropy loss denoted as L_{TFI} .

The third head is a two class classifier for PBI task. It also consists of a three-layer MLP with a softmax function. The input is the hidden representation of [CLS] token. It indicates whether the predicted text line should start a new paragraph. It has a cross entropy loss denoted as L_{PBI} .

And the last head is a sequence labeling head using the BIO format for CE task. It is implemented as a three-layer MLP with a sigmoid function. The input is the hidden representation of each text token. It can be seen as a multi-line named entity recognition problem. It has a cross entropy loss denoted as L_{CE} .

The total loss function is a weighted sum of four types of losses, calculated as Eq.(8). w_i are the relative weight of each task. Figure 5 illustrates the proposed model architecture designed to address the four tasks outlined previously.

$$Loss = w_1 * L_{ROD} + w_2 * L_{TFI} + w_3 * L_{PBI} + w_4 * L_{CE} \quad (8)$$

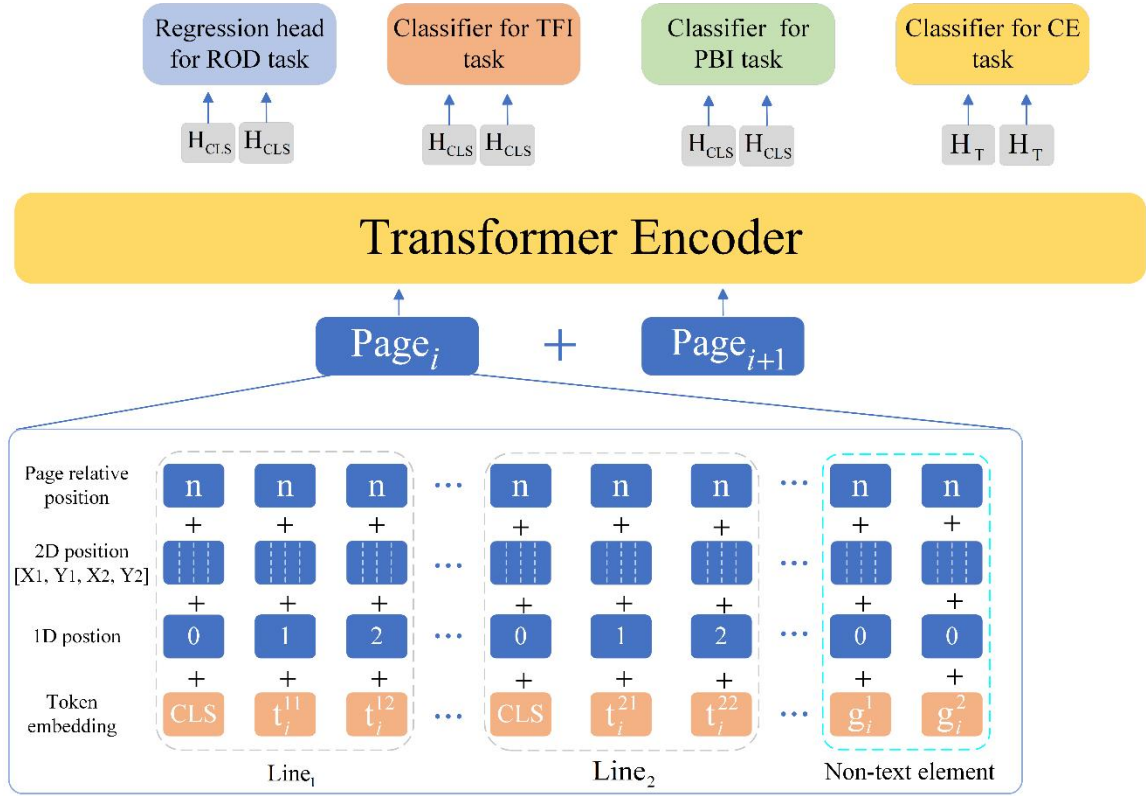


Figure 5. The network structure of ParseFormer. Each input consists of elements from two consecutive PDF pages, where each element (token) in each page is represented by four features: page relative position, 2D position $[X1, Y1, X2, Y2]$, 1D position, and token embedding.

5 Experiments and Analysis

5.1 Experiment Setup

The experiment is conducted with the following three aims. (1) Evaluate the performance of proposed method on four tasks when trained on constructed dataset. (2) Check whether the four subtasks have any mutual influence. (3) Find the optimal number of parameters of the main parsing model. In sections 5.2, 5.3, 5.4 and 5.5, we report the result of four tasks with combined training respectively. In section 5.6, we give the result of separated training and combined training with different number of parameters.

The parameters of the model were initialized with released weights of the LayoutLMv3-base [26]. After several pretraining epochs, we set the relative loss weights for the four tasks (ROD, TFI, PBI, and CE) to $[2, 1, 1, 5]$, respectively. We also explored two automatic weight adjustment methods: DWA [28] and GradNorm [29]. But the final results indicated that there was little difference in effectiveness among these methods. The page position scale is set to 20. For the 2D spatial features, the normalized coordinates are mapped to a fixed discrete range, and we set $S = 1000$ in our experiments. The training batch size is set to 8. The learning rate is set to $9e-6$.

5.2 Evaluation Metrics

For TFI, PBI and CE, we use precision, recall and F1 score as evaluation metrics. For ROD, we refer to [20], using the ARD metric. We also set up a new metric ARD^2 . Given a sequence $A = [a_1, a_2, \dots, a_n]$ and its generated reordered sequence $B = [a_{i_1}, a_{i_2}, \dots, a_{i_n}]$ where $\{i_1, i_2, \dots, i_n\} \subseteq \{1, 2, \dots, n\}$, the ARD^2 score is calculated as follows:

$$ARD^2(A, B) = \frac{1}{n} \sum_{a_k \in A} (k - I(a_k, B))^2 \quad (9)$$

where a_k is the k -th element in sequence A and $I(a_k, B)$ is the index of a_k in sequence B . n is the length of sequence A . The setup of ARD^2 is for distinguish two types of errors. The first type of error involves all elements being predicted in positions close to their correct order, with only a few minor offset-like errors. The second type of error occurs when the positions of most elements are correctly predicted, but a very small number of elements have predictions that are significantly off from their correct positions. When most errors are the second type, ARD^2 will be high. Finally, four metrics ARD , ARD^2 , Spearman correlation coefficient (Corr) and average processing time per page (seconds) are adopted to evaluate the order detection result.

5.3 Evaluation on TFI task

To evaluate the effectiveness of ParseFormer on text function identification (TFI), we compared it with three representative PDF parsing solutions:

- PPStructure [7]: Trained on the CDLA dataset [30] containing 5k Chinese academic document images. Since its label schema differs from ours, we only report results for table captions, figure captions, titles, and references.
- Marker [8]: It uses a vision based encoder-decoder model to detect the block type. Similarly, we only report its result on Table caption, Figure caption and title.
- MinerU [9]: Another vision-based method for block type detection, with results reported for table captions, figure captions, and titles.

The detailed comparison is summarized in Table 4. All three baselines rely primarily on vision-based approaches to detect block types. Overall, ParseFormer achieves F1-scores above 0.92 across all 16 main text function categories. Among the different labels, figure caption and table caption are relatively harder to identify than the others. In some PDF pages, the caption text is too close to the body text and the model tends to regard it as the body text. For these two labels, the vision-based baselines and ParseFormer achieve similar results, highlighting their intrinsic difficulty. On the other hand, vision-based methods perform worse than text-based approaches for detecting section titles, often misclassifying them as body text when there are insufficient distinct visual cues. This observation underscores the importance of leveraging semantic textual features for robust text function identification.

In addition, we separately evaluate concatenated content, a challenging subtype within the ‘‘Other type’’ category. ParseFormer achieves an F1-score of 0.79 on this subtype. This task is more challenging than identifying other text function labels, as it requires integrating multiple signals: the page number from the previous page, possible textual cues such as ‘‘上（下）接第XXX页’’, and the local text semantics near the inter-page boundary. The results indicate that the proposed method is capable of effectively combining these signals to determine whether a segment continues from the previous page or belongs to a different article. In some cases, the model correctly identifies concatenated content even when a page contained text from three different articles. Error analysis shows that most misclassifications occur in two-column layouts, where the left column is often predicted rightly as concatenated content while the right column is not. This may be because textual cues are usually located at the top-left of the left column, which is spatially distant from the right column, reducing their influence. In a few cases, the model mistakenly classified the English title and abstract as concatenated content, likely due to an unusually large blank space between the references and the English title/abstract on the page.

Table 4. Comparison results for the TFI task on the test set

Text Type	ParseFormer			MinerU [9]	Marker [8]	PPStructure [7]
	P	R	F1	F1	F1	F1
Table caption	0.96	0.87	0.92	0.94	0.49	0.90
Figure caption	0.94	0.92	0.93	0.86	0.47	0.94
Paragraph text	0.98	0.98	0.98	-	-	-

Chinese Article Title	0.97	0.94	0.96	0.82	0.82	0.83
English Article Title	0.96	0.88	0.94			
Section Title	0.94	0.93	0.93			
Chinese Abstract	0.99	0.99	0.99	-	-	-
English Abstract	0.93	0.97	0.95	-	-	-
Chinese author name	0.97	0.95	0.96	-	-	-
English author name	0.90	0.95	0.92	-	-	-
Chinese Keywords	0.99	1.00	0.99	-	-	-
English Keywords	0.92	0.95	0.93	-	-	-
Chinese Institution	0.99	0.97	0.98	-	-	-
English Institution	0.94	0.95	0.94	-	-	-
Reference	0.92	0.93	0.93	-	-	0.95
Other type	0.93	0.90	0.92	-	-	-
(concatenated content)	0.81	0.78	0.79	-	-	-
Macro avg	0.95	0.94	0.95	-	-	-

Note: Concatenated content is a subtype within the “Other type” category and is reported separately for further analysis. Macro-average scores are computed over the 16 main text function categories only. “concatenated content” is not counted as an additional category

5.4 Evaluation on PBI task

To assess the effectiveness of ParseFormer in paragraph boundary identification (PBI), we compared it with three alternative solutions:

- All-False: A naive baseline where no line is predicted to start a new paragraph;
- Marker [8]: Using the results of the layout analysis model and post-processing rules;
- MinerU [9]: Using the results of layout analysis and post-processing rules. For recognized list block or directory block, the line break is added.

The overall results are summarized in Table 5. The rule-based systems Marker and MinerU achieve considerably stronger results, with F1 scores of 0.92 and 0.94, respectively. Their performance benefits from layout analysis combined with handcrafted heuristics such as indentation, spacing, and block-level patterns, and they are particularly effective when the document formatting follows consistent typographic conventions. However, these methods also exhibit clear limitations: their reliance on visual and positional heuristics makes them vulnerable to OCR noise, heterogeneous journal layouts, and list-style structures. For example, in scanned or low-quality PDFs, indentation may be lost or distorted, and spacing irregularities may cause rules to misfire, leading to missed or spurious paragraph boundaries.

In contrast, our proposed method achieves the best overall performance, with an F1 score of 0.98 (precision 0.98, recall 0.98), substantially reducing errors compared to rule-based approaches. A key advantage of our approach is that it does not rely solely on layout cues but also incorporates semantic information from the surrounding text. This enables the model to differentiate, for instance, between a caption line that visually resembles paragraph text and a genuine paragraph boundary, or to identify list items that should be treated as separate paragraphs even when spacing cues are weak. By jointly modeling contextual semantics and positional features, the system learns to capture subtle linguistic signals (e.g., discourse markers, syntactic structures) that rules cannot cover, making boundary detection far more robust across diverse document formats.

Although the PBI task is relatively simpler compared to the other three parsing tasks, accurate detection of paragraph boundaries is critical for preserving the logical flow and discourse structure of scientific texts. Without reliable segmentation, downstream modules such as text function identification and citation extraction may operate on improperly segmented input, leading to cascading errors. The high performance of our method therefore not only demonstrates the effectiveness of combining semantic and positional cues for this task, but also provides a stable foundation for the entire multi-task parsing framework.

Table 5. Comparison results for the PBI task on test set

Method	P	R	F1
All-False	0.38	0.49	0.42
MinerU [9]	0.95	0.93	0.94
Marker [8]	0.91	0.94	0.92
ParseFormer	0.98	0.98	0.98

5.5 Evaluation on CE task

To evaluate the performance of our citation extraction (CE) module, we compared ParseFormer with several open-source large language models (LLMs), including: Qwen3-8b [31], Qwen3-14b [31], Qwen3-32b [31], and DeepSeek-v3 [32]. For a fair comparison, all LLMs were applied in a generative annotation setup, where paragraph-level inputs were used to mitigate hallucination issues caused by overly long contexts. The specific prompts employed for each model are provided in the appendix. The overall results are summarized in Table 6.

Overall, our proposed method achieves the best performance, reaching an F1 score of 0.82, which is competitive with and exceeds the baseline LLMs (0.61–0.72). Notably, our approach maintains a strong balance between precision (0.79) and recall (0.87), whereas the LLM baselines display different trade-offs. For example, Qwen3-8b achieves relatively high precision (0.79) but lower recall (0.61), suggesting that some valid citations are missed. DeepSeek-v3, although the largest model among those compared, obtains an F1 of 0.61, with higher recall (0.73) but lower precision (0.64), indicating a tendency to over-predict. Larger models such as Qwen3-32b show slight recall improvements (0.67), yet overall performance remains below our method.

Error analysis further shows that many incorrect cases in our method stem from OCR recognition errors. We currently use PaddleOCR-v4 [7] as the text detection model, and it struggles with superscript digits, small font sizes, and spacing, which can distort tokens like “et al.”. By contrast, LLMs often suffer from overgeneralization: they frequently annotate names or years mentioned in text as citations, even when no formal citation format is present. Despite being provided with explicit positive and negative examples in the prompts, this behavior persisted, suggesting that prompt-based LLM annotation lacks sufficient structural grounding. These results highlight that while LLMs possess strong language modeling capabilities, their tendency to hallucinate citation spans and their sensitivity to noisy inputs limit their reliability for CE. By leveraging shared encoder representations and explicit supervision, our multi-task framework achieves more accurate and robust extractions.

Table 6. Comparison results for the CE task

Method	P	R	F1
Qwen3-8b [31]	0.79	0.61	0.68
Qwen3-14b [31]	0.70	0.63	0.66
Qwen3-32b [31]	0.78	0.67	0.72
DeepSeek-v3 [32]	0.64	0.73	0.61
ParseFormer	0.79	0.87	0.82

5.6 Evaluation on ROD task

To evaluate the effectiveness of our proposed regression-based reading order detection module, we conducted experiments on the constructed test dataset. We compared ParseFormer with four representative baselines:

- PPstructure [7] : It sorts the layout block bboxes by adding a threshold to normal reading rule.
- Marker [8]: It uses a vision encoder decoder model to predict the order of text blocks detected by its layout analysis model.
- MinerU [9]: It combines rules and a deep reading order model to detect the reading order. It models

the reading order detection task as an element classification problem.

- Binary head: This method uses a pairwise approach as described in Section 2.2. We train a binary classifier to predict the pairwise order relationship matrix and then decode the global order from this matrix. The summation of each row in the matrix is taken to be the rank score of each text line. The higher score means higher the ranking position.

All methods were evaluated under the same setting, with inference performed on a single A100 GPU using a batch size of two pages. The overall comparison results are summarized in Table 7.

Table 7. Comparison results for the ROD task

Method	ARD	ARD ²	Corr	Avg Time Per Page (seconds)
PPstructure [7]	0.11	0.56	0.98	0.002
Marker [8]	0.20	3.85	0.99	0.114
MinerU [9]	0.07	1.26	0.99	0.046
Binary head	0.04	0.67	0.99	0.058
ParseFormer	0.03	0.63	0.99	0.047

From Table 7, it can be seen that the proposed method achieved the lowest ARD (0.039). This result can prove the effectiveness of the devised reading order detection method. The result of binary head method is close to the proposed regression method. This can validate our assumption that the multi-layer self-attention mechanism is enough for each element to perceive the position of other elements in the whole page. There is no need to further fuse the representations of each two elements in the predict head. From ARD², it can be seen that while the PPstructure [7] method is slightly lower than ours, its rule-based method are not robust, being very sensitive to the OCR detection results, slight changes can affect the final reading order especially when the page has lots of equations.

For processing efficiency, the results in Table 7 show clear trade-offs among different approaches. The rule-based PP-Structure achieves the fastest average processing time (0.002 s per page), but its ARD is much higher, reflecting limited robustness. Conversely, the generative method Marker is the slowest (0.114 s per page). Comparing the two learning-based baselines, our regression head achieves about 18% faster inference than the binary head (0.047 s vs. 0.058 s) while maintaining a slightly lower ARD. MinerU reaches the second-best speed (0.046 s per page), but its formulation as a fixed-size classification problem constrains its applicability, especially when the number of elements to be ordered is large. Overall, ParseFormer delivers a favorable balance between efficiency and accuracy.

Upon examining the model predictions on the test dataset, we observed that most error cases occurred in cases where the layout contains two columns — the left column being significantly longer than the right column. In such cases, the model often incorrectly predicted that the long portion of the left column should come after the content in the right column. This issue could potentially be addressed by augmenting the training data with more examples featuring diverse column length ratios.

5.7 Comparison between joint and separated training

To investigate whether training the four tasks on shared encoding layers would affect the performance of individual tasks, we conducted separate training for each of the four tasks. The result is shown in Table 8. We give the ARD metric for ROD task and F1 for other three tasks. As can be seen, the performance gap between separate and joint training reveals different sensitivities of the four tasks. For ROD, the difference is the most pronounced: separate training achieves an ARD of 0.031, compared to 0.039 under joint training, suggesting that reading order prediction is particularly sensitive to representation sharing and may be interfered with by other tasks. CE also shows a noticeable, though smaller, gap, with F1 decreasing from 0.84 (separated) to 0.82 (joint). By contrast, TFI and PBI remain virtually unaffected: TFI consistently achieves 0.94 F1, while PBI maintains a strong 0.98 in both settings. Overall, these results indicate that although joint training introduces some trade-offs for ROD and CE, the overall performance degradation is modest, and multi-task learning remains an efficient and practical strategy.

To further examine whether increasing the model capacity can mitigate the task interference, we scaled the encoder depth from the base 12-layer model to 16 layers (medium) and 20 layers (large). The results

show that enlarging the model yields only marginal improvements. For ROD, ARD decreases slightly from 0.039 (Base) to 0.037 (Medium/Large). CE improves modestly from 0.82 to 0.83. TFI and PBI remain unchanged, maintaining 0.94 and 0.98, respectively. These findings suggest that simply increasing the number of parameters does not substantially alleviate the inter-task interference and the performance gains are not proportional to the additional computational cost. Taken together, the results demonstrate that while separated training can provide marginal benefits for ROD and CE, the joint multi-task framework achieves a more balanced trade-off between accuracy, model efficiency, and scalability.

Table 8. Results for separated training and different-sized models

	ROD	TFI	CE	PBI
Separated training-12	0.031	0.94	0.84	0.98
Base-12	0.039	0.94	0.82	0.98
Medium-16	0.037	0.94	0.83	0.98
Large-20	0.037	0.94	0.83	0.98

6 Discussion

6.1 Implications

Our work shows that a unified multi-task learning framework can effectively support multiple fundamental subtasks of scientific PDF parsing. By jointly modeling reading order detection, text function identification, paragraph boundary identification, and citation extraction with shared encoder representations, ParseFormer provides theoretical evidence that fine-grained structural understanding can be achieved without training separate models for each objective. At the same time, the constructed Chinese scientific PDF dataset and the extensible design of ParseFormer have clear practical value: they enable large-scale transformation of Chinese scientific literature into structured formats such as JSON or Markdown and directly support downstream applications including citation network analysis, knowledge base construction, and retrieval-augmented systems. These contributions highlight both the research significance of unified structure learning in document AI and its utility in real-world scholarly scenarios.

6.2 Limitations and Future Work

Despite these contributions, several limitations remain. First, the current study focuses on four subtasks. Although they are fundamental for structured parsing, other objectives—for instance, detailed reference parsing or title hierarchy prediction—are not yet included. However, owing to the shared encoder representation of ParseFormer, the framework can be readily extended by adding lightweight task-specific heads or post-processing modules to support such objectives. Second, the framework depends on OCR outputs, which introduces errors especially on noisy or low-quality scans, constraining the upper bound of performance. Third, while the dataset spans 14 disciplines, its scale is still modest compared with large multilingual corpora, and further expansion will strengthen model generalization. Finally, the experiments are limited to LayoutLMv3 as the encoder backbone; exploring stronger multimodal backbones may further enhance performance.

7 Conclusion

In this paper, we proposed ParseFormer, a multi-task learning framework for parsing Chinese scientific journal PDFs, enabling the simultaneous execution of four key tasks—reading order detection, text function identification, paragraph boundary identification, and citation extraction—within a unified deep learning model. Experimental results demonstrate the effectiveness of sharing encoder representations across tasks, achieving a favorable balance between accuracy, robustness, and model efficiency. In addition, the proposed approach does not rely on language-specific features, making it potentially applicable to scientific PDFs in other languages as well.

Fine-grained parsing of scientific PDFs provides a solid foundation for knowledge base construction, information retrieval, and other downstream applications involving Chinese academic literature. In future work, we plan to expand our dataset to cover a broader variety of document types and introduce additional

text function labels. Beyond the four fundamental tasks addressed in this study, the unified encoder representation enables the incorporation of further parsing objectives, such as heading hierarchy recognition and reference entry parsing, thereby improving the completeness of structured document understanding. We also aim to more closely integrate the layout analysis module into the ParseFormer framework to further enhance parsing efficiency and scalability.

References

- [1] E. Petrovich *et al.*, ‘Bibliometrics beyond citations: introducing mention extraction and analysis’, *Scientometrics*, vol. 129, no. 9, pp. 5731–5768, Sep. 2024, doi: 10.1007/s11192-024-05116-x.
- [2] Yang Zhao, Zhixiong Zhang, Yufei Wang, and Xin Lin, ‘Identifying research contributions based on semantic analysis of citation sentences: A case study of the 2021 Physiology or Medicine Nobel Prize laureates’, Jul. 2023, doi: 10.5281/ZENODO.8306057.
- [3] T. L. Kalmer, C. M. F. Ancajas, Z. Cheng, A. S. Oyedele, H. L. Davis, and A. S. Walker, ‘Assessing the ability of ChatGPT to extract natural product bioactivity and biosynthesis data from publications’, Aug. 02, 2024, doi: 10.1101/2024.08.01.606186.
- [4] C. Zhang, P. Mayr, W. Lu, and Y. Zhang, ‘An editorial note on extraction and evaluation of knowledge entities from scientific documents’, *Scientometrics*, pp. s11192-024-05166–1, Oct. 2024, doi: 10.1007/s11192-024-05166-1.
- [5] W. Fan *et al.*, ‘A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models’, in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Barcelona Spain: ACM, Aug. 2024, pp. 6491–6501. doi: 10.1145/3637528.3671470.
- [6] ‘GROBID’, GitHub - kermitt2/grobid: A machine learning software for extracting information from scholarly documents. Accessed: Dec. 05, 2024. [Online]. Available: <https://github.com/kermitt2/grobid>
- [7] ‘PaddleOCR’. Github. [Online]. Available: <https://github.com/PaddlePaddle/PaddleOCR/tree/main/ppstructure>
- [8] V. Paruchuri, ‘Marker’. Accessed: Dec. 05, 2024. [Online]. Available: <https://github.com/VikParuchuri/marker>
- [9] B. Wang *et al.*, ‘MinerU: An Open-Source Solution for Precise Document Content Extraction’, Sep. 27, 2024, *arXiv*: arXiv:2409.18839. Accessed: Oct. 22, 2024. [Online]. Available: <http://arxiv.org/abs/2409.18839>
- [10] H. Wei *et al.*, ‘General OCR Theory: Towards OCR-2.0 via a Unified End-to-end Model’, Sep. 03, 2024, *arXiv*: arXiv:2409.01704. Accessed: Oct. 03, 2024. [Online]. Available: <http://arxiv.org/abs/2409.01704>
- [11] L. Blecher, G. Cucurull, T. Scialom, and R. Stojnic, ‘Nougat: Neural Optical Understanding for Academic Documents’, Aug. 25, 2023, *arXiv*: arXiv:2308.13418. Accessed: Jul. 15, 2024. [Online]. Available: <http://arxiv.org/abs/2308.13418>
- [12] Z. Li *et al.*, ‘MonkeyOCR: Document Parsing with a Structure-Recognition-Relation Triplet Paradigm’, Jun. 05, 2025, *arXiv*: arXiv:2506.05218. doi: 10.48550/arXiv.2506.05218.
- [13] A. Nassar *et al.*, ‘SmolDocling: An ultra-compact vision-language model for end-to-end multi-modal document conversion’, Mar. 14, 2025, *arXiv*: arXiv:2503.11576. doi: 10.48550/arXiv.2503.11576.
- [14] J. Poznanski *et al.*, ‘olmOCR: Unlocking Trillions of Tokens in PDFs with Vision Language Models’, Jul. 02, 2025, *arXiv*: arXiv:2502.18443. doi: 10.48550/arXiv.2502.18443.
- [15] M. Aiello and {Arnold M.W.} Smeulders, ‘Bidimensional Relations for Reading Order Detection’, in *EPRINTS-BOOK-TITLE*, University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science, 2003.
- [16] H.-Y. Ma, H.-H. Huang, and C.-L. Liu, ‘Reading between the Lines: Image-Based Order Detection in OCR for Chinese Historical Documents’, *AAAI*, vol. 38, no. 21, pp. 23808–23810, Mar. 2024, doi: 10.1609/aaai.v38i21.30572.
- [17] L. Quirós and E. Vidal, ‘Reading order detection on handwritten documents’, *Neural Comput & Applic*, vol. 34, no. 12, pp. 9593–9611, Jun. 2022, doi: 10.1007/s00521-022-06948-5.
- [18] L. Li, F. Gao, J. Bu, Y. Wang, Z. Yu, and Q. Zheng, ‘An End-to-End OCR Text Re-organization

- Sequence Learning for Rich-Text Detail Image Comprehension’, in *Computer Vision – ECCV 2020*, vol. 12370, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds, in Lecture Notes in Computer Science, vol. 12370. , Cham: Springer International Publishing, 2020, pp. 85–100. doi: 10.1007/978-3-030-58595-2_6.
- [19] L. Qiao, C. Li, Z. Cheng, Y. Xu, Y. Niu, and X. Li, ‘Reading order detection in visually-rich documents with multi-modal layout-aware relation prediction’, *Pattern Recognition*, vol. 150, p. 110314, Jun. 2024, doi: 10.1016/j.patcog.2024.110314.
- [20] Z. Wang, Y. Xu, L. Cui, J. Shang, and F. Wei, ‘LayoutReader: Pre-training of Text and Layout for Reading Order Detection’, Aug. 27, 2021, *arXiv*: arXiv:2108.11591. Accessed: Apr. 09, 2024. [Online]. Available: <http://arxiv.org/abs/2108.11591>
- [21] Z. Liu *et al.*, ‘Swin Transformer: Hierarchical Vision Transformer using Shifted Windows’, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9992–10002. doi: 10.1109/ICCV48922.2021.00986.
- [22] Z. Zhao, H. Kang, B. Wang, and C. He, ‘DocLayout-YOLO: Enhancing Document Layout Analysis through Diverse Synthetic Data and Global-to-Local Adaptive Perception’, Oct. 16, 2024, *arXiv*: arXiv:2410.12628. doi: 10.48550/arXiv.2410.12628.
- [23] E. Hetzner, ‘A simple method for citation metadata extraction using hidden markov models’, in *Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries*, in JCDL ’08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 280–284. doi: 10.1145/1378889.1378937.
- [24] B. Wellner, A. McCallum, F. Peng, and M. Hay, ‘An integrated, conditional model of information extraction and coreference with applications to citation matching’, 2012, *arXiv preprint arXiv:1207.4157*.
- [25] ‘开放获取 - 大家·维普期刊OA出版平台’. Accessed: Dec. 05, 2024. [Online]. Available: <http://oa.datalawjournal.com/openAccess>
- [26] Y. Huang, T. Lv, L. Cui, Y. Lu, and F. Wei, ‘LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking’, Jul. 19, 2022, *arXiv*: arXiv:2204.08387. Accessed: Jul. 06, 2024. [Online]. Available: <http://arxiv.org/abs/2204.08387>
- [27] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, ‘Listwise Approach to Learning to Rank - Theory and Algorithm’.
- [28] S. Liu, E. Johns, and A. J. Davison, ‘End-to-end multi-task learning with attention’, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1871–1880.
- [29] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, ‘Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks’, in *International conference on machine learning*, PMLR, 2018, pp. 794–803.
- [30] ‘CDLA: A Chinese document layout analysis (CDLA) dataset’. Accessed: Dec. 05, 2024. [Online]. Available: <https://github.com/buptlihang/CDLA>
- [31] A. Yang *et al.*, ‘Qwen3 Technical Report’, *arXiv preprint arXiv:2505.09388*, 2025.
- [32] DeepSeek-AI, ‘DeepSeek-V3 Technical Report’, 2025, *arXiv preprint arXiv:2412.19437*. [Online]. Available: <https://arxiv.org/abs/2412.19437>

Author Contributions

Guangyin Zhang: Research proposition and design, initial draft writing, model experimentation;
 Liping Gu: Paper revision, writing guidance;
 Maodi Hu: Paper revision, writing guidance, experimental guidance;
 Wang meng: Writing guidance, paper revision;
 Mengting Zhang: Paper revision, writing guidance;
 Yang Zhao: Paper revision, writing guidance;
 Zhixiong Zhang: Paper revision, writing guidance;

Acknowledgements

This study was funded by China Grant 2022YFF0711900 & China Grant 2023YFF0725600

Appendix

A. Annotation Standards and Label Definitions

To ensure the consistency and reproducibility of annotations, we established detailed standards for the 16 text function labels tailored to Chinese scientific journals. Each label was defined with clear semantic boundaries, supported by positive and negative examples during the annotation process. The main guidelines are summarized below:

- **Figure Caption:** Text associated with a figure that describes its content, usually presented in a distinct font size and separated from body text by clear whitespace. Figure-related notes are also included. Excludes paragraphs that merely reference a figure in the main text.
- **Table Caption:** Text associated with a table that describes its content, usually in a distinct font size and separated from body text by clear whitespace, including any table footnotes. Excludes paragraphs that merely reference a table in the main text.
- **Chinese Article Title:** The main title of the article in Chinese, including subtitles. Typically appears on the first page in a larger font size.
- **English Article Title:** The main title of the article in English, including subtitles. Usually follows the Chinese abstract and keywords on the first page, though in some journals it may appear on the last page after the reference section. These titles are typically translations of their Chinese counterparts.
- **Chinese Abstract:** The structured or unstructured summary section labeled as “摘要”. It usually appears after the Chinese article title and author names, and is often displayed in a font style distinct from the body text.
- **English Abstract:** The structured or unstructured summary section labeled as “Abstract”. It typically follows the English article title and author names. This abstract is usually a translation of the Chinese abstract.
- **Section Title:** Any heading indicating a new structural unit (e.g., “引言”, “背景”, “结论”, “参考文献”) in the body text.
- **Chinese Author Name:** Complete author names in Chinese, typically listed beneath the Chinese article title.
- **English Author Name:** Complete author names in English, typically listed beneath the English article title. These keywords are typically translated from the Chinese keywords.
- **Chinese Institution:** Author affiliation information written in Chinese, often indicated with superscripts or other indices.
- **English Institution:** Author affiliation information written in English, often indicated with superscripts or other indices.
- **Chinese Keywords:** A list of keywords written in Chinese, usually appearing beneath the Chinese abstract.
- **English Keywords:** A list of keywords written in English, usually appearing beneath the English abstract. These keywords are typically translated from the Chinese keywords.
- **Reference:** Bibliographic entries in the reference section at the end of the article. Footnotes containing references at the bottom of pages are excluded in this version. In-text citations are not labeled here; they are annotated separately under the Citation Extraction task.
- **Paragraph Text:** All remaining sections of the article body, including introduction, methods, results, and discussion. The first paragraph following the “Abstract” is annotated as Paragraph Text rather than as a separate “Introduction”.
- **Other type:** Any content that does not fall into the fifteen categories above. This includes marginal notes, journal headers/footers, cross-article concatenation lines (e.g., “上接第X页”), author notes, acknowledgements, funding notes, and publication metadata such as the Chinese Library Classification (CLC) code, document identification code, article serial number, and project/funding statements.

To reduce ambiguity, we created an internal annotation manual with examples for each label, and annotators were trained with it before large-scale labeling. During the iterative correction process, samples with inconsistent labeling across annotators were revisited and corrected according to these definitions.

B. Prompts used in the LLM-assisted annotation process

Prompt for citation extraction annotation

The given text segment is extracted from a Chinese academic journal PDF. It is a concatenation of many lines on the original PDF image. Inside the text segment, the special token "[BR]" represents the line breaks from the original OCR results.

Now you need to do the following tasks

identify the position of literature citation and wrap them with symbol [START] and [END], such as xxx[START][1][END]xxx, xxx[START][4-5][END]xxx, xxx[START](张三, 2022)[END]xxx.

Literature citation annotation rule:

You only need to find two types citations-number digits format and (author, year) format. Ignore other types of citations. Wrap each citation span using [START] and [END], remember don't take the "[Digit]" appeared in math formula as the citation symbol. In some papers, the "[Digit]" will follow a page number, like "[1]234", this means it refers to the page 234 of literature [1].

Examples table:

Input	**annotation Output**
'xxxx'(no citation case)	'xxxxxx'
'xxx[1]xxx'	'xxx[START-1][1][END-1]xxx'
'xxx[2-4]xxx'	'xxx[START-1][2-4][END-1]xxx'
'xxx[7,9-12]xx'	'xxx[START-1][7,9-12][END-1]xx'
'xxx[11, [BR] 24-27]xx'	'xxx[START-1][11, [BR] 24-27][END-1]xx'
'xxx[9]125xx'	'xxx[START-1][9][END-1]125xx'
'xxx[14]75xx'	'xxx[START-1][14][END-1]75xx'
'xx(Zhao et al, 2011)xxx'	'xxx([START-2]Zhao et al, 2011[END-2])xxx'
'xxx(张三, [BR] 2023)xxx'	'xxx([START-2]张三, [BR] 2023[END-2])xxx'
'xxx(李四, 2028; 王五等, 2027; Yan et al, 2023)xxx'	'xxx([START-2]李四, 2028[END-2]; [START-2]王五等, 2027[END-2]; [START-2]Yan et al, 2023[END-2])xxx'
'xxx(李四, 2028; 王五等, 2027; Yan et al, (不完整))'	'xxx([START-2]李四, 2028[END-2]; [START-2]王五等, 2027[END-2]; [START-2]Yan et al, (不完整))'
'李华等, 1980a, 1980b)xxxx'	'[START-2]李华等, 1980a, 1980b[END-2])xxxx'
'xxxxx'(no citation case) 'xxxxxx'	

Note:

1. The provided text is the result of OCR, so it might exist some small OCR errors, such as the omission of spaces. You can try to correct them.
2. The special token "[BR]" should not be omitted or modified.
3. If the provided text do not contain the citations, you still need repeat the whole provided text.
4. Do not generate any additional explanations except the output JSON.

Now i will provide you the list of text blocks, you need annotate all the citations:

Prompt for Text function type and paragraph boundary annotation-1

Task Description

The provided image is a scanned PDF page from a Chinese academic journal, containing OCR-detected bounding boxes (bboxes). Each bbox has a blue-colored number immediately to its left, serving as an identifier.

Background Information

In Chinese journal PDFs, articles typically include both Chinese and English versions of the title and abstract. Usually:

1. The **Chinese article title** and **Chinese abstract** appear first.
2. The **English article title**, **English abstract**, and **English keywords** follow.
3. The **main body text** begins after the English abstract and keywords.

Your Tasks

1. Assign an **appropriate label** to each bbox.
2. Determine whether each bbox **should start a new line**.

The label set: [

"Page header";

"Chinese article title": Article title in chinese language,

"English article title": Article title in english language,

"Chinese author": For the author names appeared under the chinese article title,

"English author": For the author names appeared under the english article title,

"Chinese institution": For the institutions appeared under the chinese article title,

"English institution": For the institutions appeared under the english article title,

"Chinese abstract": For the chinese version article abstract,

"English abstract": For the english version article abstract,

"Chinese keywords";

"English keywords";

"Section title";

"Table caption";

"Table footnote";

"Figure caption";

"Paragraph text";

"Reference": list of reference entries after the body text part,

"Page footer";

"Other type": All bbox which is not appropriate to be assigned any above label, label it as other type

]

Labeling Rules

1. **Consider multiple factors** when assigning labels, including the text content inside each bbox, its surrounding text, and its visual layout.
2. **Captions for figures and tables** should not be embedded within paragraph text.
3. If a bbox **does not clearly fit into any category**, label it as "other type".
4. "chinese article title" and "english article title" should **only** be assigned to titles that appear on the **first page of the PDF in a large font size**. If a title appears on other pages in a regular font size, classify it as "page header" or "page footer".
5. "reference" labels should **only** be assigned to reference sections that appear **after the main body text**. If similar reference text appears earlier (e.g., on the first page), it should be labeled as "page header".
6. The bbox with text "参考文献" should be given the section title label.
6. **On the first page**, elements such as "中图分类号", "文献标识码", and "DOI" should be labeled as "other type".
7. **Author introductions** that appear **after the references section** should be labeled as "other type". The label chinese author can only be given author names under the article title in first page.
8. **Figure captions** typically appear **below** figures, while **table captions** usually appear **above** tables.
9. For all the other type bboxes that you do not sure to give which label, assign them to "other type".
 - If the bbox **contains the beginning of a new reference entry** → 'True'.

Prompt for Text function type and paragraph boundary annotation-2

Special Cases

- **First Page of the PDF (Contains Chinese Title & Abstract)**
 - Assign "page header" to all text **above the article title**, including references and citation information.
 - Assign "page footer" to all text **at the bottom of the page**, which often includes funding acknowledgments, author affiliations, and submission/acceptance dates.
- **Subsequent Pages (No Large-Size Title or Chinese Abstract)**
 - Assign "page header" to elements like **page numbers, author names, and article titles** that appear at the top of the page.

New Line Rules

When determining whether a bbox should start a new line, consider both the **text content** and **visual layout**:

1. **Titles & Section Titles** should always start a new line ('True').
2. **Bounding boxes that begin a new paragraph** should start a new line ('True').
3. **For the first bbox (on the top of the image) in each column's body part (left column or right column)**:
 - If it is of type "paragraph text" and **does not** have a clear left indent → 'False'.
 - If it **has** a clear left indent → 'True'.
 - If it is of type "reference":
 - References typically follow this format: **[index number] author list, title, journal/conference name, year, volume**.
 - The **blue number** is an added identifier and should not be mistaken for the reference index number.
 - If text in the bbox **does not** start with an '[index number]' and an author list → 'False'. Incomplete means that this reference is a continuation of previous image which do not shown here or from previous column. It will be set to True in previous page or previous column, so set this line to be False in here.
 - If the bbox **contains the beginning of a new reference entry** → 'True'.
4. **Lines following an equation** **without** a left indent **should not start a new paragraph**.
 - If a line starts with "其中, XXX.", explicitly set it to 'False'.
5. **Each reference entry** in the reference list **must start on a new line**.
6. **Figure captions & Table captions**, the begin line of captions should be set to True, the continuation line should set to False. When the chinese caption and english caption are put together, take them as one caption and only set new line to true to begin line no matter it is a chinese version or english version.

Now i will provide you the list of ocr bboxes, do the three task for each OCR bbox. Generate your output in original order and Do not skip or omit any bbox.